LREC 2018 Workshop

The 13th Workshop on Asian Language Resources

PROCEEDINGS

Edited by

Kiyoaki Shirai

ISBN: 979-10-95546-24-5
EAN: 9 791095 546245

7 May 2018

Proceedings of the LREC 2018 Workshop
“The 13th Workshop on Asian Language Resources”

7 May 2018 – Miyazaki, Japan

Edited by Kiyoaki Shirai

http://www.jaist.ac.jp/project/alr13/index.html

Organising Committee

- Olivia Kwong (The Chinese University of Hong Kong)
- Kiyoaki Shirai (Japan Advanced Institute of Science and Technology)*
- Karthika Vijayan (National University of Singapore)

*: Main editor and chair of the Organising Committee

Programme Committee

- Masayuki Asahara (National Institute for Japanese Language and Linguistics)
- Minhwa Chung (Seoul National University)
- Koiti Hasida (The University of Tokyo)
- Sarmad Hussain (Center for Language Engineering, Al-Khwarizmi Institute of Computer Science, University of Engineering & Technology Lahore)
- Hansaem Kim (Yonsei University)
- Mamoru Komachi (Tokyo Metropolitan University)
- Binyang Li (University of International Relations, Beijing)
- Lianfang Liu (Guangxi Computing Center, PRC)
- Wang Lei (Institute for Infocomm Research)
- Masnizah Mohd (Universiti Kebangsaan Malaysia)
- Hammam Riza (Agency for the Assessment and Application of Technology, BPPT)
- Rachel Edita Roxas (National University, Philippine)
- Minoru Sasaki (Ibaraki University)
- Kazutaka Shimada (Kyushu Institute of Technology)
- Sanghoun Song (Incheon National University)
- Virach Sornlertlamvanich (Sirindhorn International Institute of Technology, Thammasat University)
- Takenobu Tokunaga (Tokyo Institute of Technology)
- Masao Utiyama (National Institute of Information and Communications Technology)
- Derek Fai Wong (University of Macau)

Preface

This 13th workshop on Asian Language Resources (ALR13) focuses on language resources for Asian region, which has more than 2,200 spoken languages. There are now increasing efforts to build multilingual, multi-modal language resources, with varying levels of annotations, through manual, semi-automatic and automatic approaches, as the use of Information and Communication Technology (ICT) spreads across the region. Correspondingly, the development of practical applications on these language resources has also been rapidly growing. A series of the workshop on Asian Language Resources aims to forge a better coordination and collaboration among researchers on these languages and in the Natural Language Processing (NLP) community in general, to develop common frameworks and processes for this purpose. ALR13 is supported by Asian Federation of Natural Language Processing (AFNLP), which has a dedicated Asian Language Resource Committee (ARLC), whose aim is to coordinate the important ALR initiatives with different NLP associations and conferences in Asia and other regions. This workshop consists of twelve oral papers with a wide variety of languages. Three papers present language resources (LRs) of Indo-Aryan languages (Hindi, Urdu, and Bengali), three papers present Japanese LRs, two papers present Korean LRs, two papers present Chinese LRs, one paper presents LR of Malay/Indonesian, and one paper presents Burmese LR.

Kiyoaki Shirai May 2018

Programme

9:25 – 9:30	Opening
	Session 1
9:30 – 10:00	Satomi Matsumoto, Masayuki Asahara and Setsuko Arita *Japanese clause classification annotation on the 'Balanced Corpus of Contemporary Written Japanese'*
10:00 – 10:30	Yasutomo Kimura, Yuzu Uchida and Keiichi Takamaru *Speaker Identification for Japanese Prefectural Assembly Minutes*
	Session 2
11:00 – 11:30	Jaeho Han, Changhoe Hwang, Seongyong Choi, Gwanghoon Yoo, Eric Laporte and Jeesun Nam *DECO-MWE: Building a Linguistic Resource of Korean Multiword Expressions for Feature-Based Sentiment Analysis*
11:30 – 12:00	Gwanghoon Yoo and Jeesun Nam *A Hybrid Approach to Sentiment Analysis Enhanced by Sentiment Lexicons and Polarity Shifting Devices*
12:00 – 12:30	Li Song, Yuan Wen, Sijia Ge, Bin Li, Junsheng Zhou, Weiguang Qu and Nianwen Xue *An Easier and Efficient Framework to Annotate Semantic Roles: Evidence from the Chinese AMR Corpus*
12:30 – 13:00	Hiroki Nomoto, Hannah Choi, David Moeljadi and Francis Bond *MALINDO Morph: Morphological dictionary and analyser for Malay/Indonesian*
	Session 3
14:00 – 14:30	Tayyaba Fatima, Raees Ul Islam and Muhammad Waqas Anwar *Morphological and Orthographic Challenges in Urdu Language Processing: A Review*
14:30 – 15:00	Aishwary Gupta and Manish Shrivastava *Enhancing Semantic Role Labeling in Hindi and Urdu*
15:00 – 15:30	Soumil Mandal, Sainik Kumar Mahata and Dipankar Das *Preparing Bengali-English Code-Mixed Corpus for Sentiment Analysis of Indian Languages*
15:30 – 16:00	Raoul Blin *Automatic Evaluation of Alignments without using a Gold-Corpus - Example with French-Japanese Aligned Corpora*

Session 4

16:30 – 17:00 Karthika Vijayan and Haizhou Li
Parallel Speak-Sing Corpus of English and Chinese Songs for Speech-to-Singing Voice Conversion

17:00 – 17:30 Win Win Thant and Kiyoaki Shirai
Automatic Acquisition of Opinion Words from Myanmar Facebook Movie Comments

17:30 – 17:35 Closing

Table of Contents

Japanese clause classification annotation on the 'Balanced Corpus of Contemporary Written Japanese'

Satomi Matsumoto♡, Masayuki Asahara♣, Setsuko Arita♢
♡ Ritsumeikan University
♣ National Institute for Japanese Language and Linguistics
♢ Ritsumeikan University

Abstract

Inter-clause syntactic and semantic structures are important to process semantic reasoning. This paper presents clause boundaries and class annotation on the 'Balanced Corpus of Contemporary Written Japanese'. The annotation is based on the Tori-Bank labelset, which provides the most fine-grained clause classes. We reformulated the legacy syntactic pattern into a syntactic-dependency-based pattern. Two annotators modified the automatically extracted clause boundary candidates. In this study, we investigate the patterns of disagreement in the annotation.

1. Introduction

Clause boundary detection and classification are important issues in the detection of causal and temporal relations between two events. This paper presents clause boundaries and annotations applied to the 'Balanced Corpus of Contemporary Written Japanese' (BCCWJ) (Maekawa et al., 2014); the annotation is based on the surface pattern of the morphemes. Then, the clause boundaries are categorized by their syntactic and semantic classes in Tori-Bank (Ikehara, 2007). Though the clause classes were designed for Japanese-English machine translation, the nevertheless serve as a basis for reasoning on inter-clause relations.

The original Tori-Bank patterns are based on a legacy POS tagset. We reproduced the clause patterns and adapt them to the UniDic POS tagset and word segmentation schema with Bunsetsu-based dependency structure. Then, we annotated the clause boundaries for syntactic and semantic classes on newspaper samples from the 'BCCWJ'. Next, we evaluated the syntactic or semantic classes of clause boundaries that tended to show annotation discrepancies.

2. Annotation Schema for Clause Classification

The clause boundary classification is based on the Tori-Bank schema (Ikehara, 2007). Tori-Bank is a corpus developed at Tottori University in 2007 in order to compile a Japanese semantic pattern dictionary for compound and complex sentences. The clause boundary patterns are hierarchically defined, in four layers. The top level of the classification consists of Nominal Clauses (補足節:HS), Adnominal Clauses (名詞修飾節:MS), Adverbial Clauses (副詞節:FU), and Coordinate Clauses (並列節:HR). The second level of the classification is made up of 26 classes, and the third level is made up of 52 classes. We use the third-level labels for our annotation. Below, we describe the Tori-Bank clause labels and provide annotated examples from BCCWJ.

2.1. Nominal Clauses

Nominal clauses (補足節:Hosoku-Setsu, HS) are classified into noun clauses (HSa), interrogation clauses (HSb), and quotation clauses (HSc) at the second level.

Noun clauses (HSa) are then classified at level three, into *koto* (コト) form (HSa100), *no* (ノ) form (HSa200), *tokoro* (トコロ) form (HSa300), and clause + case particle form (HSa400). Below, (1) is an example of HSa100. 'こと' *koto* appears at the end of the clause. (2) is an example of HSa200. 'の' *no* appears at the end of the clause. (3) is an example of HSa300. 'ところ' *tokoro* appears at the end of the clause. These words are relative pronouns. (4) is the example of HS400. This is a zero (relative) pronoun before the case particle "に".

(1) 制度を 育て上げることが ぜひとも 必要。
seido-wo sodateagerukoto-ga zehitomo hitsuyou.
'It is really needed to develop the system.'
HSa100 (*no*),
[BCCWJ Sample ID: PN1c_00001]

(2) 赤字額が 最も 多い の は
akajigaku-ga mottomo ooi no-ha
東京都大江戸線の 三百十一億円
toukyouooedosen-no sanbyakujuuichiokuen
だった
datta
'Oedo Line in the Tokyo Metropolitan had the largest deficits at 31.1 billion yen.'
HSa200 (*koto*), [PN2e_00001]

(3) ボールが けれる ところ まで 回復している
booru-ga kereru tokoro-made kaifukushiteiru
'I have recovered to the point of being able to kick a ball.'
HSa300 (*tokoro*), [PN2f_00002]

(4) 実務レベルで 協議する に とどまっている
jitsumureberu-de kyougisuru-ni todomatteiru
'Stop at the negotiation at the practical level.'
HSa400 (zero pronoun),
[PN3a_00002]

Interrogation clauses (HSb) are subclassified into alternative/choice questions (HSb100), in (5), and question with interrogative words (HSb200), in (6).

(5) 復調の めどがたっているの か 表情は、
fucchouno medogatatteiruno-ka hyoujou-ha
明るかった。
akarukatta

'Perhaps due to the increase in the possibility of recovery, his facial expression brightened.'
`HSb100 (alternative/choice question), [PN2f_00002]`

(6) 減税額が 実際に どの程度 違うの か
genzeigaku-ga jissai-ni donoteido chigauno-ka
検証してみよう。
kenshoushitemiyou

'Let ' s verify to what extent the amount of tax reduction actually differs. '
`HSb200 (question with interrogative word), [PN4c_00002]`

Quotation clauses (HSc) are classified into direct quotations (HSc100), as in (7), and indirect quotations (HSc200), as in (8). We classified them by the presence or absence of quotation marks, '「」'.

(7) 「... 卒業証書は 出す」と 言った
"... sotugyoushousho-ha dasu"-to itta

' "··· I will submit your graduation certificate," he said.'
`HSc100 (direct quotation), [PN4g_00003]`

(8) 目立つ 生徒だったと いう。
medatsu seitodata toiu

'It is said that he was a prominent student.'
`HSc200 (indirect quotation), [PN1c_00001]`

2.2. Adnominal Clauses

Adnominal clauses (名詞修飾節: Meishishushoku Setsu, MS) are classified into relative clauses (MSa), apposition clauses (MSb), and clauses with contractive expressions (MSc), clause with functional expressions (MSd), and clause with collocational expressions (MSe).
Relative clauses (MSa) are those in which the modifiee is an argument of the subordinate clause's end predicate. Relative clauses are classified into restrictive (MSa100), as in (9), and non-restrictive (MSa200), as in (10). These two then each subdiscriminated by whether the modifiee is a normal noun (MSa100) or a proper noun (MSa200).

(9) 呼び出して 注意する 先生も いたが
yobidashite chuuisuru sensei-mo ita-ga

'Although there were also teachers who summoned and warned students.'
`MSa100 (relative, restrictive), [PN1c_00001]`

(10) この日は 腰の 重い 安芸乃島に
konohi-ha koshi-no homoi akinoshima-ni
快勝。
kaishou.

'Achieved an easy win over Akinoshima, who was slow to act, today.'
`MSa200 (relative, non-restrictive), [PN1e_00004]`

Apposition clauses (MSb) is that the modifiee has an appositive relation with the clause.

(11) 低迷する 日本経済の 「負の側面」を
teimeisuru nihonkeizai-no "funosokumen"-wo
象徴する 結果に なった
shouchousuru kekka-ni natta

'Became results that symbolized the " negative sides " of the sluggish economy of Japan.'
`MSb (apposition), [PN1e_00001]`

Contractive adnominal clauses (MSc) are adnominal clauses that are neither relative nor apposition clauses.

(12) 試合は １点を 争う 展開。
shiai-ha itten-wo arasou tenkai

'The match developed into a competition for one point.'
`MSc (contractive), [PN1e_00003]`

Functional adnominal expressions (MSd) are pairings of an adnomial clause and a modifee to express a functional meaning. They are subclassified into functional adnominal expressions with relative pronouns (MSd100), as in (13), functional sentence-end expressions (MSd200), as in (14), idiomatic expressions (MSd300), as in (15), and functional adnominal expressions in adverbial usage, as in (16).

(13) サラダと 聞いて
sarada-to kiite
思い浮かべた ものは、野菜サラダ
omoiukabetamonoha, yasaisarada

'When I heard the word "salad," vegetable salad came into my mind.'
`MSd100 (functional adnomial expression with relative pronoun), [PN1a_00002]`

(14) 廃止の 理由は、 授業時間を
haishi-no riyuu-ha, jugyoujikan-wo
確保する ため
kakuhosurutame

'The reason for the abolition is to secure class hours.'
`MSd200 (functional sentence end expression), [PN1a_00002]`

(15) 何かに 没頭できる という 点では、
nanika-ni bottoudekiru toiu tendeha,

'In terms of being able to be absorbed in something,'
`MSd300 (idiomatic expression), [PN3b_00004]`

(16) 焦げ付きを 懐具合に 見合った 範囲に
kogetsuki-wo futokoroguai-ni miatta han'i-ni
抑えたい
osaetai

'I wish to limit bad debts to a range commensurate with my financial standing.'

MSd400 (functional adnominal expression in adverbial usage), [PN1g_00002]

Example (17) shows a collocational expression with the pattern 'predicative + conjunctive + の *no*'.

(17) ずっと 入院したままの 例も 珍しく
zutto nyuuinshitamama-no rei-mo mezurashiku
なかった
nakatta
'Cases where the patient stayed hospitalized for a long time were not rare.'
MSe (predicative + conjunctive + の *no*), [PN3a_00003]

2.3. Adverbial Clauses

The adverbial clause (副詞節: Fukushi Setsu, FU) is classified by semantic features.

First, (FUa) are temporal clauses, indicating a time-point or the duration of an event, as in (18). The, (FUb) are causal clauses, indicating a cause or result, as in (19).

(18) バックが 暗い中、ストロボの 光が 鳥に
bakku-ga kurainaka sutorobo-no hikari-ga tori-ni
集光して、
shukou-shite,
'Against a dark background, concentrate the stroboscope on the bird,'
FUa100 (temporal), [PN1d_00001]

(19) 新聞で 報道され、逃げ切れないと 思って
shinbun-de houdousare nigekirenai-to omotte
自首した
jishushita
'He surrendered as he thought he would not be able to escape after being featured in the news.'
FUb100 (causal), [PN1f_00002]

Next, (FUc) are conditional clauses, subclassifiable into nomothetic conditionals (FUc100), as in (20), accidental conditionals (FUc200), as in (21), and imaginary conditionals (FUc300), as in (22).

(20) 世界から 見れば、一地方大学。
sekai-kara mire-ba ichichihoudaigaku.
'A regional university from the perspective of the world.'
FUc100 (nomothetic conditional), [PN3b_00001]

(21) 自分が 注意を したら
jibun-ga chuui-wo shitara
逃げ出したことなどを
nigedashitakotonado-o
'It would not have escaped, if you had paid attention.'
FUc200 (accidental conditional) [PN1b_00003]

(22) 1人に 力点を 置くなら、断食や
hitori-ni rikiten-wo oku-nara, danjiki-ya
ダイエットには 格好の 状況と
daietto-ni-ha kakkou-no joukyou-to
言えるだろう
ierudarou
'It could be a situation suitable for fasting and dieting, if the emphasis is put on one person.'
FUc300 (imaginary conditional), [PN3b_00004]

Subsequently, (FUd100) captures attendant circumstances, as in (23), and (FUd200) is aspectual clauses, as in (24). These two are under the 'attendant circumustances' label at the second level label (FUd).

(23) 実効性を 考慮して 慎重に
jikkousei-wo kouryoshite shinchou-ni
決めるべきだ
kimerubekida
'One should decide cautiously, while taking effectiveness into consideration.'
FUd100 (attendant circumstances), [PN3g_00001]

(24) 今と 同じように、子どもの
ima-to onajiyou-ni, kodomo-no
家庭環境を 把握する
kateikankyou-wo haakusuru
'Figure out the children's family environment, just like now.'
FUd200 (aspectual), [PN1a_00002]

The, (FUe) are contrastive clauses, as in (25); (FUf) are objective clauses including necessity and intention, as in (26); and (FUg) conveys degree of action or state, as in (27).

(25) ■■さんは 病院に 運ばれたが、
XX-san-ha byoouinn-ni hakobareta-ga,
間もなく 死亡した
mamonaku shiboushita
'Mr. X was sent to the hospital, but he died shortly afterwards.'
FUe (contrastive), [PN4f_00001]

(26) 不良債権の 最終処理という
furyousaiken-no saishushoritoiu
外科大手術を するには 大量の
gekadaishujutu-wo suru-ni-ha tairyou-no
輸血が 必要で
yuketu-ga hitsuyou-de
'Massive blood transfusion is necessary for major surgeries, which are regarded as the final treatment of bad debts.'
FUf (objective), [PN1b_00004]

(27) 日本は アメリカに 言われるまでもなく、
nihon-ha amerika-ni iwarerumademonaku,
国内経済の 安定を 第一に 考えて
kokunaikeizai-no antei-wo daiichi-ni kangae-te

‘Needless to be pointed out by the USA, Japan first considers the stability of domestic economy.’
FUg (degree), [PN1b_00004]

(28) 基本合意の 覚書を 交わした上で、
kihongoui-no oboegaki-wo kawashitaue-de,
合弁会社を 設立
goubengaisha-wo setsuritsu
‘Established a joint corporation upon the exchange of a memorandum of understanding.’
FUh (presuppositional), [PN1g_00002]

In addition, (FUi), as in (29), are means clauses; (FUj), as in (30), are dyadic or binary relation clauses; and (FUk), as in (27), are correlative clauses.

(29) 年末には 上司と 部下が 話し合って
nenmatsu-ni-ha joushi-to buka-ga hanashiatte
次期の 個人目標を つくる
jiki-no kojinmokuhyou-wo tsukuru
‘At the end of the year, the superior and the subordinate have a talk and draft personal goals for the next term.’
FUi (means), [PN5b_00003]

(30) ガスで 作るより スープの 味も まろやか
gasu-de tsukuru-yori suupu-no aji-mo maroyaka
‘The soup has a mellower taste than one made using a stove.’
FUj (dyadic), [PN4a_00001]

(31) でも じっと 眺めているうち、 怖いと
demo jitto nagameteiruuchi, kowai-to
感じ始めた
kanjihajimeta
‘But as he kept staring at it, he started to feel scared.’
FUk (correlative), [PN2b_00002]

(32) 東署で 強盗事件とみて 行方を
higashisho-de goutoujikentomite yukue-wo
追っている
otteiru
‘At the East Police Station, it was regarded as a robbery and tracking was ongoing.’
FUl (conclusive), [PN1f_00002]

Further, (FUm), as in (33), are scenery clauses; (FUn), as in (34), are presuppositional clauses; and (FUo), as in (35), are absolute clauses.

(33) 欧米のような 基盤が ない中、 市民の
oubeinoyouna kiban-ga nainaka shimin-no
実質的な 参加が 得られるように
jisshitsuteki-na sanka-ga erareruyou-ni
‘In order to gain real participation of citizens, while having no foundation like the Western countries,’
FUm (scenery), [PN3g_00001]

(34) 酒を 飲ませない 以外は、 同様の
sake-wo nomasenai igai-ha douyou-no
扱い。
atsukai.
‘The same treatment, except the prohibition to drink alcohol.’
FUn (restrictive), [PN4g_00003]

(35) 無理して 頑張る 必要は ないが、
murishite ganbaru hitsuyou-ha nai-ga,
私は 布団や カーテンなどの大物を
watashi-ha futon-ya kaatennado-no
洗濯するのが 好き
oomono-wo sentakusurunoga suki
‘Although there is no need to push myself too hard, I like washing large objects such as mattresses and curtains.’
FUo (absolute), [PN3b_00004]

Finally, (FUp) covers other adverbial clauses: (FUp100), as in (36), are functional (auxiliary verbal) expressions; (FUp200), as in (37), are idiomatic expressions; and (FUp300), as in (38), are adverbial phrases (not clauses).

(36) 医療の 倫理を 逸脱した 行為と
iryou-no rinri-wo itsudatsu-shita koui-to
いわざるを えないだろう
iwazaru-wo enaidarou
‘It is definitely an act that deviates from medical ethics.’
FUp100 (functional), [PN2b_00002]

(37) なりふり 構わず 資金を
narifurikamawazu shikin-wo
調達しようとした
choutatsushiyoutoshita
‘Tried to raise funds by fair means or foul.’
FUp200 (other, idiomatic), [PN3b_00001]

(38) 50 歳を 過ぎて なぜか エステサロンに 来た
50sai-wo sugi-te naze-ka esutesaron-ni kita
‘Somehow, I came to the esthetic salon after I had passed the age of 50.’
FUp300 (other, adverbial), [PN1b_00003]

2.4. Coordinate Clauses

Coordinate clauses (並列節; Heiretsu Setsu, HS) are classified into resultative (HRa) and contrastive (HRb) at the second level. The resultative clauses are (sub)classified again into exhaustive list (HRa100), as in (39); exemplification (HRa200), as in (40); accumulation (HRa300), as in (41); parallels (HRa400), as in (42), and negation coordination (HRa500), as in (43). An example of a contrastive clause (HRb) is (44).

(39) 年利 3% 借入れ、 三十年の
nenri 3% kariire, 30nen-no
元利均等返済方式で 返済する場合
ganrikintouhensaihoushiki-de hensaisurubaai
‘The circumstance of repaying the debt with an annual interest of 3% through a 30-year level-payment plan.’
HRa100 (exhaustive listing), [PN4c_00002]

(40) カラスよけの 糸を 張り、 ひなの ための
karasuyoke-no ito-wo hari, hina-no tame-no
筒形シェルターを 置くなど、
tsutsugatasherutaa-wo okunado,
恒久的な 営巣地に するため
koukyuuteki-na eisouchi-ni suru-tame

‘In order to make it a permanent nesting place, tie crow-repelling string and set up a cylindrical shelter for chicks.’
`HRa200 (exemplification), [PN5b_00002]`

(41) 病気の パターンごとの
byouki-no pataangoto-no
入院日数だけでなく、 医療費の
nyuuinnissuudakedenaku, iryouhi-no
全国平均値も 示された
zenkokuheikinchi-mo shimesareta

‘Shown were not only the number of hospitalization days by the pattern of diseases, but also the national average medical fees.’
`HRa300 (accumulation), [PN3a_00003]`

(42) 条約締結国に 国内の
jouyakuteiketukoku-ni kokunai-no
無形文化遺産の 保護や 目録の
mukeibunkaisan-no hogo-ya mokuroku-no
作成を 求めるとともに、
sakuse-wo motomerutotomo-ni,
国際協力のための 基金設置などを
kokusaikyourokunotameno kikinsecchinadowo
盛り込んでいる。
morikondeiru.

‘Incorporated the establishment of funds for international cooperation, along with the requests for State Parties to protect domestic intangible cultural heritage and produce catalogues.’
`HRa400 (parallel), [PN4g_00001]`

(43) 義務ではなく、 各学校の 判断で
gimudehanaku, kakugakkou-no handan-de
行われる。
okonawareru.

‘It is not an obligation and is carried out at the discretion of each school.’
`HRa500 (with negation), [PN1a_00002]`

(44) 昨日は あなたに ほほ笑んだけれど、 今日は
kinou-ha anata-ni hohoenda-keredo, kyou-ha
さようならを 言わなければいけない。
sayounara-wo iwanakerebaikenai.

‘I smiled at you yesterday, but today I have to say goodbye.’
`HRb (contrastive), [PN2b_00003]`

3. Annotation Procedures

3.1. Overview of Procedures and Target Data

We annotated third-level clause boundary labels for 52 classes on 54 BCCWJ newspaper core data A samples. The sentence boundaries, word segmentation, morphological information, *bunsetsu* (Base phrase), and *bunsetsu*-based syntactic dependency were annotated precedingly. The data consisted of 2,543 sentences and 56,922 morphemes.

The annotation procedure was based on the modification of the automatically extracted clause boundaries. First, clause boundary candidates were extracted using clause patterns with a fourth level of labels. Patterns were defined based on the morphological information and syntactic dependency relations.

3.2. Patterns of Clauses

The original Tori-Bank pattern files were provided through a contract with a data distribution organization; the specification document is available as a PDF file on the website[1]. The original patterns were based on the surface forms of the morphological analyser outputs. We reimplemented the patterns and adapted them for the UniDic POS set, lemma information, and syntactic dependency[2]. The patterns were based on a syntactic dependency structure[3]. Note, the morphological information was manually annotated on the original BCCWJ. The syntactic dependency structure was also annotated on the BCCWJ (Asahara and Matsumoto, 2016).

3.3. Annotation

First, two annotators checked the labeled clause boundary candidates based on the patterns. The first 14 files of the total of 54 files were used in a training phase.

Second, one annotator resolved the inconsistency between the two annotations.

4. Data Statistics

Table 1: Disagreement of Clause Position

	Nom	Adnom	Adv	Coord
Disagreement	102 (20%)	275 (33%)	207 (30%)	47 (24%)
Total (the final)	486	836	701	199

Table 2: Discrepancies of Clause Labels Between Two Annotators

	Nom	Adnom	Adv	Coord
Nominal	8	4	7	8
Adnominal	11	179	4	0
Adverbial	12	6	85	125
Coordinate	1	0	26	2

4.1. Statistics of the Completed Data

The table 5 presents the basic statistics of the completed data in the first and second levels. The 2,543 sentences include 673 Nominal Clauses, 1,103 Adnominal Clauses, 969 Adverbial Clauses, and 293 Coordinate Clauses.

[1] `http://unicorn.ike.tottori-u.ac.jp/toribank/data_list.html`

[2] `https://github.com/X/clause_pattern`

[3] `https://taku910.github.io/cabocha/`

(45) 雪舟作と 伝えられる 花鳥図屏風は、 10点余りが 知られている。
Sesshusakuto tsutaerareru kachouzubyoubu-ha, 10tenamari-ga shirareteiru.
'Around 10 folding screens of flower and bird by *Sesshu* were identified.'
`MSa100 (relative, restrictive) vs MSa200 (non-relative, restrictive), [PN2b_00002]`

(46) 警察当局が 危険人物と 認定した 九百三十二人に対し、
keisatsutoukyoku-ga kikenjinbutsu-to ninteishita 932nin-nitaishi,
'For 932 people who are regarded as dangerous by the police'
`MSa100 (relative, restrictive) vs MSa200 (non-relative, restrictive), [PN2c_00002]`

(47) 再建計画に 数値基準を 設けた 中間報告の 中核的な 考えに 反映されている。
saikenkeikaku-ni suuchikijun-wo mouketa chuukanhoukoku-no chuukakuteki-na kangae-ni haneisareteiru.
'.. are reflected in the core idea of the interim report (in which/that) is set as the numerical criterion for the restructuring plan'
`MSa100 (relative) vs MSb (apposition), [PN1g_00002]`

(48) 他派閥からも 引き抜いて 三十人から 五十人の 新派閥を つくることが できるんだ
tahabatsu-kara-mo hikinuite 30nin-kara 50nin-no shinhabatsu-wo tsukurukoto-ga dekirunda
'We can create a new faction with 30–50 people by hiring from the other factions'
`FUi (means) vs FUb (causal), [PN2e_00002]`

(49) 各政権の 積み残しを 一手に引き受けて、そのすべてを 処理するという . . .
kakuseiken-no tsuminokoshi-wo ittenihikiukete, sonosubete-wo shorisurutoiu . . .
'(the new government) took charge of the goods left by the previous governments and processed all of them ... '
`FUd (attendant circumstances) vs FUb (causal), [PN1b_00004]`

Table 3: Frequent Discrepancies of Clause Labels in Adnominal Clauses

	Annotator A	Annotator B
46	Relative Clause (Restrictive)	Relative Clause (Non-Restrictive)
27	Relative Clause (Non-Restrictive)	Relative Clause (Restrictive)
23	Apposition Clause	Relative Clause (Restrictive)
14	Adnominal Clause (w/ Contractive)	Apposition Clause
14	Adnominal Clause (w/ Contractive)	Relative Clause (Restrictive)
10	Relative Clause (Restrictive)	Apposition Clause

Table 4: Frequent Discrepancies among Clause Labels in Adverbial Clauses

	Annotator A	Annotator B
33	Means	Causal
12	Attendant Circumstances	Causal

The most frequent type of clause in the second level of nominal clauses was the quotation clause (342). The quotation clauses were marked with a 'と' (*to*) marker in reported speech. The frequency of noun clauses was 300. The Japanese noun clauses were marked 'の'(*no*), 'こと' (*koto*), and 'ところ'(*tokoro*).

The adnominal clauses are classified into relative clauses, apposition clauses, and others, including functional or collocational clauses. The major difference between the relative clauses and apposition clauses is whether the predicate in the clause modifier and the modified noun have predicate-argument relations.

The adverbial clauses were semantically classified into 16 classes in the second level. The most frequent type was causal relations (243). The second most frequent type was attendant circumstances (118).
Finally, the coordinate clauses were classified into the following: resultative (282) and contrastive (11) clauses.

4.2. Disagreement in the Annotation Phases

We investigated disagreements between the two annotators in the first annotation phases. We present only disagreements after the training phase, that is, in the files 15-54.
Table 1 shows disagreement on boundary detection. These disagreements were seen most frequently on adnominal clauses. Because of that, Japanese subject nominal phrases tend to be omitted, and any attributive adjective can become an adnominal clause. We introduced clauses composed of more than one *bunsetsu*. However, the judgments of the two annotators tended to disagree. We refined the definition of the clause based on the existence of a complement for the attributive adjective predicate.
Table 2 shows discrepancies in third-level labels on agreed segments.
The most frequent discrepancies were in the second-level labels in the adnominal clauses. Table 3 shows discrepancies in the third-level labels within adnominal clauses.
It is important in English clause classification to distinguish

Table 5: Second Level Labels

Label	Description	Count
HS: Nominal Clause		671
HSa	Noun	300
HSb	Interrogation	29
HSc	Quotation	342
MS: Adnominal Clause		1103
MSa	Relative	677
MSb	Apposition	213
MSc	Other	122
MSd	Functional	66
MSe	Collocational	25
FU: Adverbial Clause		969
FUa	Temporal	76
FUb	Causal	243
FUc	Conditional, Concessive	96
FUd	Attendant Circumstances	118
FUe	Contrastive	98
FUf	Objective	43
FUg	Degree	3
FUh	Presuppositional	8
FUi	Means	94
FUj	Dyadic	18
FUk	Correlative	6
FUl	Conclusive	18
FUm	Scenery	2
FUn	Restrictive	3
FUo	Absolute	68
FUp	Other	75
HR: Coordinate Clause		293
HRa	Resultative	282
HRb	Contrastive	11

between relative clauses that are restrictive and those that are non-restrictive. Whereas restrictive relative clauses of normal nouns, non-restrictive ones modify proper nouns. In contrast, in Japanese grammar the distinction between these two is vague and not overtly marked. Examples (45) and (46) show disagreeing judgments on which relative clauses were restrictive or non-restrictive. For example, the 花鳥図屏風 ‘folding screens of flower and bird’ in (45) and 九百三十二人 ‘932 people’ in (46) are difficult to specify based on world knowledge.

Moreover, the difference between relative clauses and apposition clauses is vague in the Japanese language, because the subject and object of the predicate can be omitted. Example (47) shows disagreeing judgments between relative and apposition clauses: whereas annotator A regards the example as a restrictive relative clause with the subject 中間報告 ‘the interim report’, annotator B regards it as an apposition relative clause with subject ellipsis. The sentence is too vague to resolve the attachment ambiguity.

The second-most frequent discrepancy is between coordinate and adverbial clauses. This is because the coordinate structure is a syntactic meta-structure, in which coordinate clauses are subcategorized into adverbial clauses in a clause boundary definition (Maruyama et al., 2016).

The third-most frequent discrepancies are within adverbial clauses. Table 4 shows frequent discrepancies within adverbial clauses.

The conjunctive postposition て (*te*) form in Japanese has ambiguities for semantic classification. (48) shows the discrepancy between means and causal relations. 引き抜いて ‘hiring’ can serve as both the means and the cause for 新派閥をつくる ‘create a new faction’. Then, (49) shows the discrepancy between attendant circumstances and causal relations.

4.3. Data release

These discrepancies were resolved in the second phase of checking. One annotator resolved annotation ambiguity through introspection. It was found that most disagreements were caused by oversight.

Label disagreement was caused by homographical patterns, such as in suspended form (連用中止 in Japanese) and て (*te*) form. The annotator of the second check defined a standard for differentiating them. For example, it was determined whether the mutual substitution of these types of clauses could preserve the meaning of the original sentence in a language test and thus resolve the ambiguity between coordinate and adverbial clauses. However, there are also truly ambiguous examples, which cannot be resolved even using contextual information. We put some special notes on examples that may have interpretations other than the classes with which they are annotated.

The final annotation data are available for users of the BCCWJ DVD Edition, published by the NINJAL official[4].

5. Conclusions

We present annotation data on Japanese clause boundaries with syntactic and semantic labels. We reimplemented the Tori-Bank clause patterns in the UniDic POS tagset and syntactic dependency structures. Two annotators modified the clause candidates yielded by the pattern-based analysers, and we explored the segments and labels on which they disagreed and resolved the disagreements.

The clause classes in Tori-Bank were originally designed for machine translation from English to Japanese. Some clause classes relate to for English-specific structures or issues. In our future work, we will refine the Tori-Bank clause class standard for the Japanese language. For example, adverbial clauses can be subcategorized into statement clauses and logical clauses.

Acknowledgments

This work was supported by JSPS KAKENHI Grants Numbers JP15K12888 and JP17H00917 and is a project of the Center for Corpus Development, NINJAL.

6. Bibliographical References

Asahara, M. and Matsumoto, Y. (2016). BCCWJ-DepPara: A Syntactic Annotation Treebank on the ‘Balanced Corpus of Contemporary Written Japanese’. In *Proceedings of the 12th Workshop on Asian Langauge Resources (ALR12)*, pages 49–58.

[4] NINJAL provides services to distribute BCCWJ derived data.

Ikehara, S. (2007). Japanese semantic pattern dictionary – compound and complex sentence eds. – . `http://unicorn.ike.tottori-u.ac.jp/toribank/`.

Maekawa, K., Yamazaki, M., Ogiso, T., Maruyama, T., Ogura, H., Kashino, W., Koiso, H., Yamaguchi, M., Tanaka, M., and Den, Y. (2014). Balanced Corpus of Contemporary Written Japanese. *Language Resources and Evaluation*, 48:345–371.

Maruyama, T., Sato, S., and Natsume, K. (2016). Gendai Nihongo ni-okeru Setsu no Bunruitaikei ni-tsuite, On the clause classification in Contempprary Japanese, (in Japanese). In *22nd Annual Meeting of Gengoshori-gakkai*, pages 1113–1116.

Speaker Identification for Japanese Prefectural Assembly Minutes

Yasutomo Kimura[1,2], **Yuzu Uchida**[3], **Keiichi Takamaru**[4]
[1]Otaru University of Commerce, [2]RIKEN AIP, Japan,
[3]Hokkai-Gakuen University, [4]Utsunomiya Kyowa University
kimura@res.otaru-uc.ac.jp, yuzu@hgu.jp, takamaru@kyowa-u.ac.jp

Abstract

Recently, we have been creating a corpus of Japanese prefectural assembly minutes. The corpus contains assembly minutes of all 47 prefectures between April 2011 and March 2015. This four-year period represents one term of office for the assembly members in most prefectures. In prefectural assembly minutes, the name of the speakers can be recorded in several different ways such as Japanese Hiragana and Katakana characters, and Chinese characters. Our purpose of this study is to uniquely identify the speakers by hand, reconciling the different representations of their names. This paper describes how we annotated a Japanese political corpus with speaker identity information.

Keywords: Political documents, local assembly minutes, speaker identification

1. Introduction

Many local autonomies in Japan provide open access to a variety of political documents via their websites. These documents include basic urban development plans, local assembly minutes, and ordinances. Such information obtained through internet can be used to compare local autonomies and identify the individual characteristics of the autonomies. Local assembly minutes are crucial in determining such characteristics because they include various representatives' positions on the policies enforced by that body.

Some studies to analyze local assembly minutes have been conducted by political scientists (Masuda, 2012). However, these studies have raised issues with the analysis methods of such minutes. One issue concerns the different ways in which these minutes are released to the public. There are 47 prefectures and several cities, towns, and villages in Japan; local assembly minutes are made available in a variety of ways. Gathering local assembly minutes and presenting the data collected in a unified format for analysis at a national level is therefore expensive. In this paper, we aim to create a corpus of Japanese local assembly minutes. Our overall objective is to develop a corpus that can be used for a broad range of interdisciplinary research.

Our goal is to create a prefectural assembly corpus with that is both accurate and complete. To do this, we identify which speaker has made each statement by hand.

The issue we would like to solve.

- The issue is that speakers' names are presented in several different ways. For example, in the Chiba prefecture, a local assembly member is recorded as "Pretty Nagashima" (a commonly used name) in the minutes even though his real name is "Kaoru Kataoka". In such case, we cannot recognize the speaker in the local assembly. Different expressions arise for several reasons, such as when professional names or old-style characters are used, or if there is any typographical errors. In Section 3, we will explain these speaker identification difficulties in detail. Correct speaker identification is important for conducting statistical surveys involving local assembly minutes.

Why is this topic interesting?

- Once we have identified each speaker, we will be able to answer the following three questions.

Q1 How many assembly members spoke in each Japanese prefectural assembly?

Table 1 shows the number of registered assembly members in the 47 Japanese prefectural assemblies for the period between April 2011 and March 2015. We classified the speakers into three categorizes: 25-44 years old (young), 45-64 years old (middle-aged), and 65-84 years old (old). The age composition of the assembly members during this period was as follows: 598 people were 25-44 years old, 1,790 people were 45-64 years old, 482 people were 65-84 years old, and 76 people were of unknown age. Table 2 shows the numbers of assembly members who were mentioned in the minutes as speakers, while Table 3 shows the numbers of assembly members who were not mentioned in the minutes as speakers. Table 4 divides the members who were mentioned in the minutes as speakers into the following categories: "governor", "vice-governor", "chairperson", "member" and "other (governor's agent)."

Table 1: Total number of members listed for the 47 Japanese prefectural assemblies for the period between April 2011 and March 2015.

Age	25-44	45-64	65-84	Unknown	Total
Male	547	1,616	457	71	2,691
Female	51	174	25	5	255
Total	598	1,790	482	76	2,946

Q2 Is it possible to identify differences in speech content based on gender? Figure 1 shows the mean frequencies of particular keywords per person by gender, which are normalized using the keyword frequencies for men. This shows that women often used keywords such as "tax increase", "poverty", "consumption tax"

Table 2: Number of assembly members who were listed in the minutes as speakers.

Age	25-44	45-64	65-84	Unknown	Total
Male	538	1,539	396	70	2,543
Female	50	172	23	5	250
Total	588	1,711	419	75	2,793

Table 3: Number of assembly members who were NOT listed in the minutes as speakers

Age	25-44	45-64	65-84	Unknown	Total
Male	9	7	61	1	148
Female	1	2	2	0	5
Total	10	9	63	1	153

and "pension," while men tended to use keywords such as "tourism", "expressway" and "decentralization."

Q3 Is it possible to clarify the difference of the speech content depending on age? Figure 2 shows the mean frequencies of particular keywords per person by generation, which are normalized using the keyword frequencies for the middle aged (45-60 years old) group. Young members (25-44 years old) often used keywords such as "internet" and "children wait-listed". Middle aged members (45-64 years old) made many remarks involving keywords such as "nursing care" and "medical expenses". Old members (65-84 years old) tended to use keywords such as "regional revitalization" and "self-defense forces".

Again, we note that it is important to identify the speakers in the local assembly minutes. Only after annotating the minutes with speaker information we can answer the above three questions accurately and completely.

Why could not previous research solve this issue?

1. No other corpus of local assembly minutes which is both accurate and complete exists.
2. Previous research has not analyzed the differences between the 47 prefectural governments.

Our contributions can be summarized as follows.

- We create a political corpus that includes everything that was said in the prefectural assemblies; we are also careful in ensuring that is both accurate and complete (Table 2).
- We identify all the expression ways in which each speaker's name is expressed (Section. 3).
- We identify characteristic statistical distributions using our corpus (Figure 1 and Figure 2).
- We create a training dataset for speaker identification.

Table 4: Classification of the speakers in the assemblies.

Governor	Vice-governor	Chairperson	Member	Other
60	156	376	2,793	3,076

2. Creating the corpus

In this section, we used to create our corpus of Japanese prefectural assembly minutes.

2.1. Prefectural assembly minutes

Japan is divided into 47 prefectures. The corpus of prefectural assembly minutes provides a language resource which clearly indicates who spoke in the assembly, as well as when, where and what they said. Figure 3 shows an example of assembly minutes from the Yamagata prefecture. There are a lot of 1,788 local governments throughout Japan, including prefectures, cities, towns, villages, and special wards in Tokyo. Approximately 86% (of these1,542 municipalities) publish their local assembly minutes on the internet. Minutes are created for several types of public local assembly, including plenary assemblies (regular and extraordinary meetings) and committees (e.g., budget and audit committees). In this paper, we focus on minutes for regular prefectural meetings.
We have collected minutes from the regular meetings in all 47 prefectures that were held between April 2011 and March 2015. This four-year period represents one term of office for the assembly members in most prefectures. The total amount of collected text data is about 1.8 GB.

Table 5: Number of local assembly members by type.

Prefecture	City	Ward	Town &Village	Total
2,687	18,654	902	11,271	33,514

3. Speaker identification

We have identified all the speakers who were listed in the Japanese prefectural assembly minutes. In this section, we discuss the ways in which we classified and identified the speakers.

3.1. Speaker type

We classified the speakers as follows: "governor","vice-governor","chairperson", "member" and "other (governor's agent)." Table 4 shows the number of speakers of each type.Note that the number of "other" speakers is larger than the total number of "members".

3.2. Identifying the speakers

As mentioned in the previous section, the speakers include not only just politicians but also other people, such as governor's agent. The speakers' names are also expressed in different ways in the local assembly minutes. Furthermore,

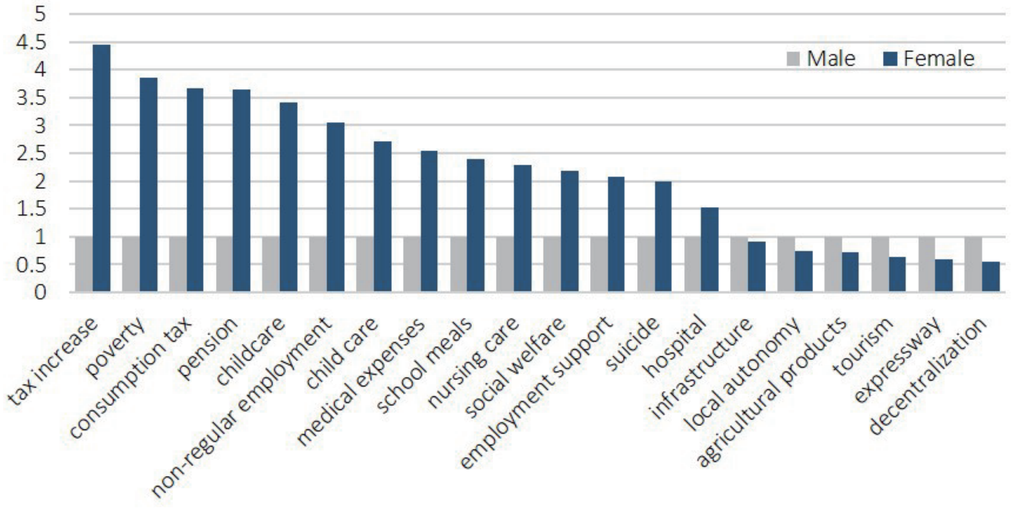


Figure 1: **Keyword appearance rates by gender, normalized using the male results.**

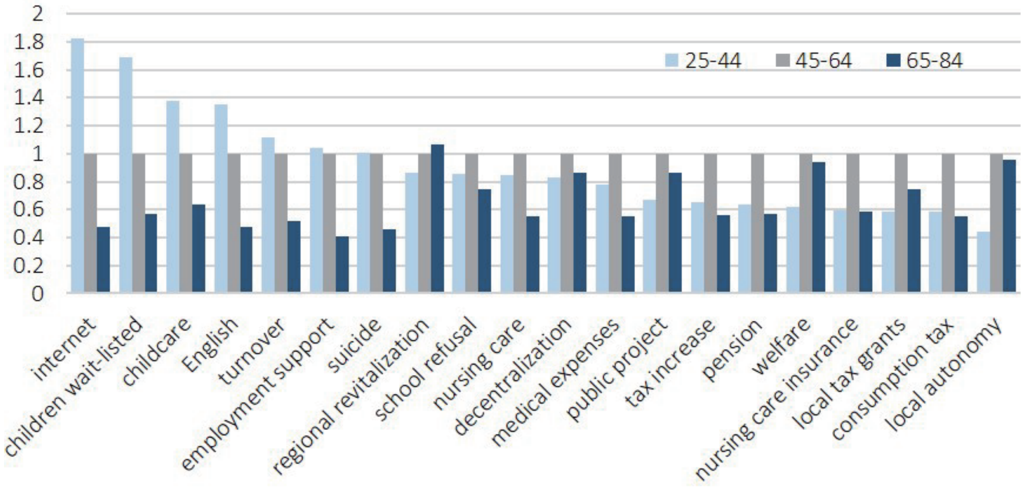


Figure 2: **Keyword appearance rates by generations, normalized using the results for the middle aged group.**

these name-expression patterns are different for each municipality, and they also include typographical errors.

In this section, we discuss the reasons why names are expressed in different ways in the minutes. We also annotate the speakers to identify them in the minutes based on the election information given on local government websites.

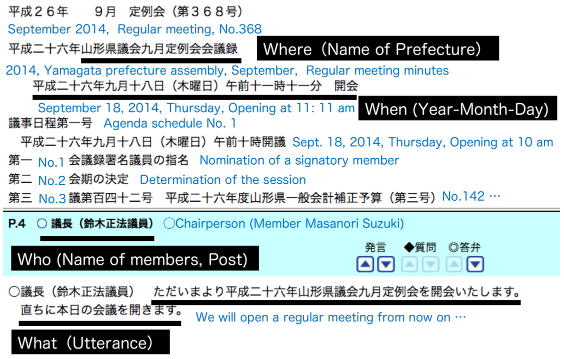
Identification can be difficult when the names are expressed differently in the minutes. The reasons for these discrepancies are as follows: the name has been written using Japanese syllabary characters; the name is a professional one; old-style characters have been used; or there is a typographical error either in the local assembly minutes or in the election information website. Japanese name can write both Chinese character and Japanese syllabary characters. Some Chinese characters used in Japanese names have old-style characters. We can change them alternately. Therefore, "Japanese syllabary characters" and "old-style characters" are unique problems on Japanese language. We have classified the discrepancies in terms of these five factors, and Table 6 shows the number of examples in each category. Note that several of these factors may contribute to a single discrepancy. Now, we discuss these five contributing factors, and we describe each one using an example.

3.2.1. Names written using Japanese syllabary characters

Commonly-used names written using Japanese syllabary characters may give rise to discrepancies when the last or first name is written using different characters, such Hiragana or Katakana characters. For example, one local assembly member is recorded as "Inamoto" using Japanese

Table 6: Number of discrepancy examples by category.

Japanese syllabary characters	Professional names	Old-style characters	Error in minutes	Error on the website	Total
447	3	186	1	39	663



平成２６年　　９月　定例会（第３６８号）
September 2014, Regular meeting, No.368
平成二十六年山形県議会九月定例会会議録
2014, Yamagata prefecture assembly, September, Regular meeting minutes
平成二十六年九月十八日（木曜日）午前十一時十一分　開会
September 18, 2014, Thursday, Opening at 11: 11 am
議事日程第一号　Agenda schedule No. 1
平成二十六年九月十八日（木曜日）午前十時開議 Sept. 18, 2014, Thursday, Opening at 10 am
第一 No.1 会議録署名議員の指名　Nomination of a signatory member
第二 No.2 会期の決定　Determination of the session
第三 No.3 議第百四十二号　平成二十六年度山形県一般会計補正予算（第三号）No.142 …
P.4 ○ 議長（鈴木正法議員） ○Chairperson (Member Masanori Suzuki)
○議長（鈴木正法議員）　ただいまより平成二十六年山形県議会九月定例会を開会いたします。
直ちに本日の会議を開きます。 We will open a regular meeting from now on …

Figure 3: **Example of assembly minutes from the Yamagata prefecture, as posted online.**

syllabary characters in the minutes; however, the member has been mentioned as “Inamoto” using Chinese characters on the election website.

3.2.2. Professional names

When members use professional names such as pen names or stage names, this can give rise to discrepancies. These professional names cannot be identified without creating a correspondence table between the real and professional names. For example, one local assembly member is recorded as “Pretty Nagashima” (his professional name) in the minutes but is mentioned as “Kaoru Kataoka” (his real name) on the election website.

3.2.3. Names written using old-style characters (Kanji variants)

Using old-style characters in one place and the current one in another place causes discrepancy and confusion. For example, one local assembly member is recorded in the minutes as “Sakae 榮” using the old-style characters but is mentioned in the election website as “Sakae 栄” using the current characters.

3.2.4. Typographical error in local assembly minutes

Typographical errors in the local assembly minutes can lead to discrepancies. For example, one local assembly member is recorded as “Ogiwara 荻原” (using a similar Chinese character) in the minutes but is mentioned as “Hagiwara 萩原” (a similar Chinese characters) on the election website.

3.2.5. Typographical errors on the election website

Typographical errors on the election website can also lead to discrepancies. For example, one local assembly member is recorded as “Shin-ichiro 真一朗” (using a similar Chinese character) in the minutes but is mentioned as “Shin-ichiro 真一郎” (using a Chinese character) on the election website.

4. Related work

The speaker identification work is an important issue in information science. Speaker identification has three major tasks as follows: (i) Speaker identification in the story such as novel and children’s story (Iosif et al., 2016)(he et al., 2013), (ii) speakers clustering for speech recognition systems (Ahmed et al., 2017), and (iii) identification of speaker’s name in several different ways. Our speaker identification is to resolve (iii) identification of speaker’s name in several different ways. For example, identification of different expressions for a speaker’s name is an significant task in NDL(National Diet Library). NDL in Japan provides Web NDL Authorities as the authority control system in Japan[1].

Recently, some studies have explored document analysis, sentiment analysis, and political debates from a political viewpoint (Yano et al., 2009; Chambers et al., 2015; Cano-Basave et al., 2016). These studies used various document datasets as political corpora. In this section, we describe corpora that include political information.

Political document analysis studies have employed various document-collections methods (such as blogs (Yano et al., 2009)) on the web; probabilistic models have been proposed for generating both blog posts and comments on blog sites. Hassanali et al. (2010) proposed a technique for automatically tagging political blog posts using support vector machines and named-entity recognition. They used blog documents as a corpus. Chambers et al. (2015) modeled sentiment analysis in the social sciences using Twitter data (over two billion tweets) as corpus. Lerman et al. (2008) automatically predicted the impact of news on the public perception about the political candidates using daily newspaper articles as corpus. Cano-Basave et al. (2016) used semantic frames to model argumentation in speaker discourse. Their presidential political debates corpus comprises 20 debates that took place between May 2011 and February 2012. Iyyer et al. (2014) applied a recursive neural network framework to detect political positions. They performed experiments using the dataset of Congressional debates and an original political dataset as a corpus. As mentioned above, political corpora typically comprise blogs, Twitter data, newspaper articles, and original political-document datasets. Our political corpus, constructed from local assembly minutes, is therefore a novel and valuable source of political information.

5. Conclusion

In this paper, we have described the annotation process we used to identify the speakers in Japanese prefectural assembly minutes. We focused on the minutes from regular meetings in the 47 Japanese prefectures between April

[1] https://id.ndl.go.jp/auth/ndla

2011 and March 2015. This four-year period represents one term of office for the assembly members in most prefectures. The speakers' name were recorded in a variety of ways such as using professional names, Japanese Hiragana and Katakana, and Chinese characters. Taking this into account, we have created a corpus of prefectural assembly minutes that is both accurate and complete. We have been publishing the website for the corpus as follows: http://local-politics.jp/47pref.

Acknowledgments
This work was supported by JSPS KAKENHI Grant Number JP16H02912 and JP17K02739.

6. Bibliographical References

Ahmed, Hany and Elaraby, Mohamed and M. Mousa, Abdullah and Elhosiny, Mostafa and Abdou, Sherif and Rashwan, Mohsen, An Unsupervised Speaker Clustering Technique based on SOM and I-vectors for Speech Recognition Systems, Proceedings of the Third Arabic Natural Language Processing Workshop, Association for Computational Linguistics, pp. 79–83, 2017.

Cano-Basave, Amparo Elizabeth and He, Yulan. *A Study of the Impact of Persuasive Argumentation in Political Debates*, Proceedings of NAACL-HLT, pp.1405–1413, 2016.

Chambers, N., Bowen, V., Genco, E., Tian, X., Young, E., Harihara, G., and Yang, E. *Identifying political sentiment between nation states with social media*, Proceedings of EMNLP, pp. 65–75, 2015.

Hassanali, Khairun-nisa, and Vasileios Hatzivassiloglou, *Automatic detection of tags for political blogs.*, Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics in a World of Social Media. Association for Computational Linguistics. pp. 21–22, 2010.

He, Hua and Barbosa, Denilson and Kondrak, Grzegorz, Identification of Speakers in Novels, Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp.1312–1320, 2013.

Iosif, Elias and Mishra, Taniya, From Speaker Identification to Affective Analysis: A Multi-Step System for Analyzing Children's Stories, Proceedings of the 3rd Workshop on Computational Linguistics for Literature (CLFL), Association for Computational Linguistics, pp.40–49, 2014.

Iyyer, Mohit and Enns, Peter and Boyd-Graber, Jordan and Resnik, Philip. *Political ideology detection using recursive neural networks*, Proceedings of the Association for Computational Linguistics, 2014.

Lerman, Kevin and Gilder, Ari and Dredze, Mark and Pereira, Fernando. Association for Computational Linguistics. *Reading the markets: Forecasting public opinion of political candidates by news analysis*, Proceedings of the 22nd International Conference on Computational Linguistics Vol. 1, pp.473–480, 2008.

Masuda, Tadashi, *Text Mining Analysis on the Minutes of Local Assemblies - A Case Study on the Takasaki City Assembly - (in Japanese)* . Takasaki City University Economics, Vol. 15, No.1, pp. 17–31, 2012.

Salton, Gerard and Buckley, Christopher, *Term-weighting approaches in automatic text retrieval*, Information processing & management Vol. 24, No 5, pp. 513–523, 1988.

Yano, Tae and Cohen, William W and Smith, Noah A. *Predicting response to political blog posts with topic models*, Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pp.477–485, 2009.

Yano, Tae and Cohen, William W and Smith, Noah A. *Predicting response to political blog posts with topic models*, Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pp.477–485, 2009.

DECO-MWE: Building a Linguistic Resource of Korean Multiword Expressions for Feature-Based Sentiment Analysis

Jaeho Han, Changhoe Hwang, Seongyong Choi, Gwanghoon Yoo, Eric Laporte*& Jeesun Nam
DICORA, Department of Linguistics and Cognitive Science,
Hankuk University of Foreign Studies, Korea
* Université Paris-Est, LIGM, CNRS, UPEM, ESIEE, ENPC, France
hanjaeho0308@gmail.com, hch8357@naver.com, csy@hufs.ac.kr, rhkdgns2008@naver.com, namjs@hufs.ac.kr
*eric.laporte@univ-paris-est.fr

Abstract

This paper aims to construct a linguistic resource of Korean Multiword Expressions for Feature-Based Sentiment Analysis (FBSA): DECO-MWE. Dealing with multiword expressions (MWEs) has been a critical issue in FBSA since many constructs reveal lexical idiosyncrasy. To construct linguistic resources of sentiment MWEs efficiently, we utilize the Local Grammar Graph (LGG) methodology: DECO-MWE is formalized as a Finite-State Transducer that represents lexical-syntactic restrictions on MWEs. In this study, we built a corpus of cosmetics review texts, which show particularly frequent occurrences of MWEs. Based on an empirical examination of the corpus, four types of MWEs have been discerned. The DECO-MWE thus covers the following four categories: Standard Polarity MWEs (SMWEs), Domain-Dependent Polarity MWEs (DMWEs), Compound Named Entity MWEs (EMWEs) and Compound Feature MWEs (FMWEs). The retrieval performance of the DECO-MWE shows 0.806 f-measure in the test corpus. This study brings a two-fold outcome: first, a sizeable general-purpose polarity MWE lexicon, which may be broadly used in FBSA; second, a finite-state methodology adopted in this study to treat domain-dependent MWEs such as idiosyncratic polarity expressions, named entity expressions or feature expressions, and which may be reused in describing linguistic properties of other corpus domains.

Keywords: Multiword Expression, Feature-Based Sentiment Analysis, DECO-MWE, Local Grammar Graph

1. Introduction

This study presents a linguistic resource of Korean Multiword Expressions for Feature-Based Sentiment Analysis (FBSA): DECO-MWE. A Recursive Transition Network methodology called Local-Grammar Graphs (Gross 1997, 1999) is adapted to construct the resources: they are compiled into Finite State Automata and Finite State Transducers and coupled with the DECO Korean Electronic dictionary that provides a diversity of linguistic information such as morphological, syntactic, semantic and sentiment-polarity information (Nam 2012, 2015). DECO-MWE covers 4 types of MWEs: Standard Polarity MWEs (SMWEs), Domain-dependent Polarity MWEs (DMWEs), Compound Named Entity MWEs (EMWEs) and Compound Feature MWEs (FMWEs). Due to the difficulties in processing the idiosyncrasy of MWEs, MWEs need to be empirically described in resources effectively structured for automatic processing. Moreover, since the Korean language shows extremely complex morphological characteristics, the resources should reliably recognize all inflectional variations of MWEs. This paper will discuss an effective way to build a linguistic resource of Korean MWEs for FBSA.

Sentiment Analysis (SA) mainly focuses on the classification of Semantic Orientation (SO), commonly referred to as polarity. Two general approaches to SA are known: the lexicon-based approach that computes the polarity value of texts through sentiment lexicons containing polarity values, and the machine-learning approach that classifies polarity by mathematical algorithms trained on a sentiment dataset (Liu 2015). With lexicon-based methods, the performance of SA fundamentally depends on the quality and size of sentiment lexicons because this approach is deeply grounded in the keyword-based vectorization of sentiment-related vocabulary (Kim et al., 2009). As compared to document- or sentence-level classification, it has become apparent that Feature-based Sentiment Analysis performs finer-grained analysis (Liu 2012).

Most of current studies on SA are grounded in machine-learning approaches such as maximum entropy, SVM or Naïve Bayes classification (Pang et al., 2002). However, conventional methods show severe limitations for FBSA, which requires processing lexical and syntactic properties. In case of inflectional linguistic phenomena in the data, frequencies of MWEs carrying SO should be considered as well. Lexicon-based approaches are more suitable to deal with a variety of MWEs for FBSA since they make it possible to analyze and calculate lexical information on sentiment words, named entities and feature nouns.

For FBSA, Liu (2012) introduces the Sentiment Quintuple model, consisting of an Entity (e), Aspect (a), Sentiment Value (s), Opinion Holder (h), and Time (t). The FBSA approach to sentiment computation requires a sentiment-annotated corpus, annotated at the level of tokens. (Hu and Liu 2006). This corpus provides a diversity of sentiment-related information and determines the reliability of the analysis. In the case of single-word expressions, explicit sentiment values can be assigned with a standard dictionary, but MWEs pose complex problems in the implementation of FBSA, since the process cannot rely on the compositionality of single words' senses. For example, *buy something for a song* means ‘buy something for a low price’ and is unrelated to singing, which manifests the idiosyncrasy of MWEs. Another type of MWEs includes compound nouns such as *anti-aging cream* that should be properly recognized for the correct analysis of the target of evaluation. Such MWEs should not be processed as several words but tokenized as one unit, and this one should be placed in the same category as *cream* in FBSA. Therefore, MWEs should be properly chunked, tagged and lemmatized in the tokenizing phase for a reliable FBSA.

The following examples in Korean show MWEs expressing polarity (i.e. (a) (b)) and analyzable as named entities (i.e. (c) and (d)):

(1a) 눈길을 끌다/ *nwunkil(eyes)-eul kkeulta(attract)*
"to attract one's attention"

(1b) 마음이 가다/ *maeum(mind)-i kata (go)*
"to catch one's fancy"

(1c) 모이스쳐라이징 크림/*moiseuchyeolaicing kheulim*
"moisturizing cream"

(1d) 컬픽스 마스카라/ *kheolpikseu maseukhala*
"curl fix mascara"

'눈길/*nwunkil'* in (a) means 'eyes', and '마음/*maeum'* in (b) 'mind'. In (a) and (b), the expressions are about attracting attention and interest respectively, not moving eyes or mind. Furthermore, (c) indicates a 'cream' the function of which is 'moisturizing' and (d) a product 'mascara' having a curl-fixing function.

As mentioned above, they need to be chunked and tagged as a single unit each and associated with all inflectional variations in order to be properly analyzed. Given that Korean is an agglutinative language in which a word is generally composed of several morphemes, this aspect should be carefully considered in case of MWEs, which makes the processing of Korean MWEs more difficult than that of English ones.

Nonetheless, as is the case with English, the majority of Korean compound words are right-headed, exhibiting 'Modifier-Head' structure (Bejček and Pavel 2010). Therefore, nominal MWEs should be grouped into semantic categories according to their heads. For example, '래스트 파운데이션/ *layseutheu phawunteisyeon'* (long-lasting foundation) refers to some kind of '파운데이션/ *phawunteisyeon*' (foundation). Besides, among Named entity MWEs, English loanwords frequently occur, which disturbs an effective recognition of MWEs. Let us consider:

(2a) 래스트 파운데이션/ *layseutheu phawunteisyeon*

(2b) 라스트 파운데이션/ *laseutheu phawunteisyeon*

(2c) 라스트 파데/ *laeseutheu phatei*

Since the transliteration of English loanwords is not strictly standardized in actual user-generated texts and loanwords are frequently abbreviated, a set of orthographic variations is observed, especially in nominal MWEs: they need to be recognized and normalized.

In this paper, we introduce the methodology of the construction of the linguistic resource DECO-MWE, in particular, based on a corpus of cosmetics review texts. The procedure used in this study is reproduced in the study of other domain corpora in Feature-based Sentiment Analysis. In Section 2, related work is briefly reviewed. In Section 3, the methodology adopted in this study is described, and in Section 4, four types of MWEs constructed in this study are discussed. A short evaluation of our linguistic resources is presented in Section 5, followed by the conclusion in Section 6.

2. Related Work

As a type of MWEs, idiomatic expressions are new units where compositionality is not observed. Since many of them may exhibit polarity values, it is important to take them into account in sentiment analysis (Williams et al. 2015). De Marneffe et al. (2008) point out that the words that constitute MWEs are combined into a single expression with a meaning independent of the individual constituents. In other words, the expression is represented within a speaker's mental lexicon just like a single word (Jackendoff 1997). In addition, the problem of analyzing text semantics can be exacerbated by the scalar variability of the expressions, as the study of Piao et al. (2003) reveals. Therefore, more attention needs to be drawn to computational MWE processing.

Baldwin and Kim (2010) define Compound Nominalization as a combination of two or more nouns into an MWE. In the perspective of feature-based sentiment analysis, named entities or feature nouns can appear in the form of compound nouns, which adds weight to the necessity to process them. Tanaka and Baldwin (2003) and Lapata and Lascarides (2003) concretely examine compound nouns and their components based on the British National Corpus (Burnard, 2000). Tanaka and Baldwin (2003) find that NN compound nouns actually cover 1.44% of the corpus . In most studies, machine learning is applied for MWE processing since it requires significant time and cost to construct sizeable linguistic resources manually.

Taboada et al. (2011) take into account MWE processing for sentiment analysis. Their study is grounded in a careful examination of sentiment linguistic resources, focusing on lexicon-based sentiment analysis of English texts using a Sentiment Orientation Calculator (SO-CAL). However, only a handful of MWEs (177 entries) such as phrasal verbs (*fall apart*) are involved in the analysis. Williams et al. (2015) emphasize the importance of MWEs as sentiment expressions and process them for sentiment analysis by making use of regular expressions to chunk them as a means of tokenization. This study is relevant in that it concentrates on the role of MWE processing in sentiment analysis, but the coverage of MWEs is limited to 580 entries, which are somewhat far from the practical usage in communication, as they have been extracted from an educational website. Besides, the chunking method based on regular expressions does not scale up to a morphologically complex language such as Korean, since the highly complex and accurate regular expressions required to process inflected forms would be difficult to modify and expand.

Most of the studies on processing MWEs for sentiment analysis are mainly conducted for English, with relatively little attention to other languages. Especially, it is difficult to process Korean MWEs because a Korean word should be analyzed into several morphemes, namely, lexical items combined with various inflectional postpositions (Kim and Shin, 2013). It means that some elements of an MWE can be followed by multiple inflectional suffixes such as nominal postpositions (in Korean, *josa*) or verbal/ adjectival postpositions (*eomi*). Moreover, it may be troublesome to capture the boundary of a word because spacing rules are not strictly respected in unstructured texts (Lee 2001).

Considering the limitations of current studies on MWE processing, this paper concentrates on constructing linguistic resources for properly recognizing and extracting Korean MWEs in the FBSA approach.

3. Methodology for Construction of DECO-MWE Resources

3.1 Data Collection

The rapid growth of Korean cosmetic industry positioned Korea as the tenth biggest market worldwide with its estimated market value of $7,427 million US dollar in 2015 (Kim 2017), leading to an increased demand for fine-grained SA.

To explore sentiment MWEs in cosmetics reviews, we crawled reviews and online User-Generated Contents, and extracted 31,506 cosmetic product names and 468 brand names from a Korean cosmetics review website called *Powder-Room*[1]. The review data is made of 796,689 tokens and 56,354 sentences.

To collect MWEs representing sentiment polarity, we first divided all sentences into two groups: sentences with occurrences of polarity words registered in DECO-Lex (Nam 2015) as sentiment words (QX- tags), and sentences without these sentiment words. This was done by applying DECO-Lex to the corpus with the Unitex platform (Paumier 2003). Then, from the second group of sentences, the most frequent neutral words (i.e. words that are assigned a tag of context-dependent polarity (QXDE) in DECO-Lex entries) were selected by computing the term-frequency table, and their concordances were generated. Our assumption was that the frequent tokens exhibiting no polarity may have a considerable role in the composition of Polarity-MWEs.

For the MWEs of named entities, 31,506 cosmetic product names and 468 brand names were examined to predict the syntagmatic combination of sequences. For the MWEs denoting features, we focused on some types of frequent words such as the equivalents of ‘color’, ‘ingredient’, ‘scent’, selecting them by consulting the sub-menus of the website.

Figure 1 describes how to construct the DECO-MWE resources systematically. After extracting and sorting the MWEs as described above, we utilized the Local Grammar Graph (LGG) formalism (Gross 1997, 1999), represented linguistic patterns in LGGs and compiled the LGGs into Finite-State Transducers (FST) through the Unitex platform (Paumier 2003). There is a coupling between the LGGs and DECO-Lex, as the LGGs use the lexical information stored in the dictionary.

The LGGs, DECO-Lex and DECO-Tagset represent syntactically complex patterns elegantly and enable correct tokenization of morphologically complicated sequences. As mentioned above, these resources associate Korean MWEs with a notably complex set of inflectional variations. Such complexity is a typological property of agglutinative languages. The DECO dictionary provides the information of all possible combinations of nouns and *josa* postpositions as well as those of predicates and *eomi* postpositions. When DECO-Lex is applied to the corpus for morphological analysis, the lexical information registered in DECO-Lex is recognized automatically by LGGs, which makes it possible to tokenize MWEs and normalize them with the canonical description. The benefits of utilizing Local Grammar Graphs may be summarized as follows:

- The flexibility of MWE processing, with the possibility of specifying input in various forms: phonemes, syllables or words.
- Compatibility with DECO-Lex and DECO-Tagset to use part of speech(POS), lemma and surface forms.
- Expressive power for lexico-syntactic irregularities and normalization.

3.2 Overview

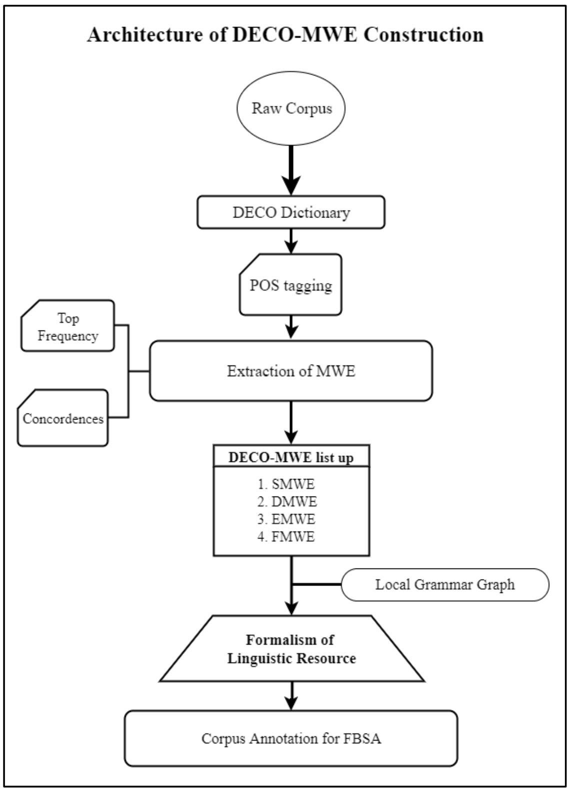


Figure 1: Architecture of DECO-MWE construction

4. The DECO-MWE Resources

DECO-MWE covers 4 types of MWEs: Standard Polarity MWEs (SMWEs), Domain-dependent Polarity MWEs (DMWEs), Compound Named Entity MWEs (EMWEs) and Compound Feature MWEs (FMWEs).

4.1 Polarity MWEs

Polarity MWEs are the most important keywords of all MWEs for FBSA. Given that MWEs are lexical units that consist of more than one word delimited by white-space (Sag et al., 2002), Polarity MWEs represent polarity expressions with unpredictable meaning. For example, *That’s a rip off* means that something is too expensive for what it is, and does not refer to an actual theft. We define Polarity MWEs as MWEs holding polarity values (i.e. expressive positive or negative opinion) in FBSA.

We categorized Polarity MWEs in types according to their syntactic structures. There are three types of combination of nouns and predicates:

- Noun + Noun
- Noun + Predicate

[1] Korean cosmetic review website: www.powderroom.co.kr

- Predicate + Predicate (including *josa* and *eomi* since each expression can be inflected in various ways)
- ETC (Idiosyncratic sequences)

We constructed LGGs based on the examination of lexical-syntactic patterns observed in the cosmetics domain. Since the interpretation or properties of a significant amount of MWEs vary according to various domains, it seems more effective to divide these MWEs into two sub-categories: Standard Polarity MWE (SMWE) and Domain-dependent Polarity MWE (DMWE).

Notice that the lexicon-based approach to SA aims to tokenize multiword expressions having semantic orientation. Therefore, we assign polarity values in the form of the DECO-PolClass tagset (Nam 2015) and the tokenization is performed by chunking and tagging MWEs with the LGGs.

4.1.1 Standard Polarity MWEs (SMWEs)

SMWEs convey a unique polarity value regardless of sub-domains. An example of SMWEs is '바가지를 쓰다/*pakaci-leul sseuta*'(-) (similar to the English expressions *cost an arm and a leg, cost a fortune, pay through the nose*).

In order to construct a sizable quality SMWE resource, we considered not only the 82 SMWEs extracted from the cosmetic corpus but the 205 lists proposed by existing research (Kim, 2000) as well. Additionally, we supplemented the resource by listing up meticulously selected 834 SMWEs from the extensive data crawled from the web-based idiom dictionary [2]. This complementary approach makes the SMWE cover extensive idiomatic MWEs holding polarity values quantitatively as well as qualitatively. The SMWEs were formalized in LGGs.

Once a reliable list of MWEs is prepared, LGGs representing these MWEs are manually constructed under a form of the directed graph as follows:

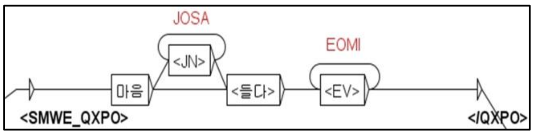


Figure 2. An example of Positive SMWE

The LGG in Figure 2 represents certain MWEs such as '마음에 들다/*maeum-ey teulta*' ('to catch one's fancy'). In this LGG, units without '<' and '>' represent surface forms (e.g. 마음), while those with < > such as <들다> represent lemma forms. As a matter of fact, <들다> recognizes a verb root when it is followed by a certain number of inflectional suffixes recognized by the <EV> symbol. By chunking the expressions and enclosing them in XML-like tags such as <SMWE_QXPO> and </QXPO>, this LGG tags automatically SMWEs in accordance with their corresponding semantic orientation.

The overall figures of SMWEs are classified by polarity orientation as shown below.

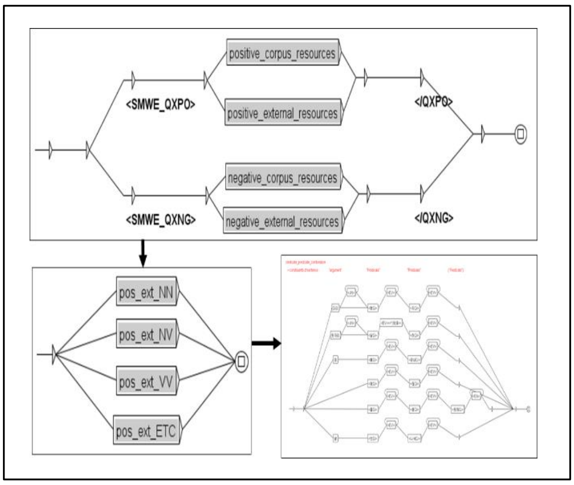

positive_corpus_resources
positive_external_resources
negative_corpus_resources
negative_external_resources
<SMWE_QXPO>
</QXPO>
<SMWE_QXNG>
</QXNG>
pos_ext_NN
pos_ext_NV
pos_ext_VV
pos_ext_ETC

Figure 3: Overall SMWE LGG excerpt

Figure 3 illustrates the main LGG which contains the sub-graphs (e.g. Positive SMWE sub-graphs) to process extensive data of SMWEs. The LGGs representing SMWEs in this study include 1,121 types.

4.1.2 Domain-Dependent Polarity MWEs (DMWEs)

Let us consider:

(3a) 촉촉하게 스며들다/*chokchokhagey seumyeo teulta* ("something soaked into skin moistly" (+))

(3b) 모공 부각/*mokong pwukak* ("skin pore expansion"(-))

(3c) 빛이 나다/*pich-i nata'* ("shine on the skin" (+))

Certain Polarity MWEs extracted from the corpus such as (3) convey domain-dependent polarities, thus they are classified as DMWEs: they belong to a specific domain where each MWE shows its own unique meaning. In terms of semantic properties, they may be classified as figurative expressions, having a vital role in conveying positive or negative opinion. Overall, distinguishing SMWEs and DMWEs has a practical advantage in extending polarity MWEs for FBSA. Figure 4 is an example of DMWEs:

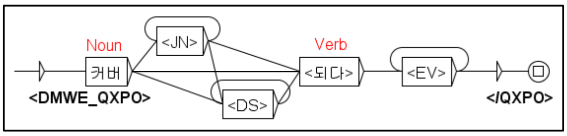


Figure 4: A LGG of Positive DMWE

This LGG represents certain sequences such as '커버가 되다/*kheopeo-ka toyta*'("perfect cover makeup(+)") that conveys a positive opinion without any explicit sentiment word. The verb <되다> is inflected by the postposition recognized by the <EV> symbol, and the noun '커버/*kheopeo*' is followed by one or several postpositions recognized by <JN>. The possibility of adverbial modification is marked by the <DS> symbol recognizing the insertion of possible adverbial words such as '잘/*cal* (well)' or '완전히/*wanceonhi* (completely)'.

[2] NAVER Idiom Dictionary: krdic.naver.com/list.nhn?kind=idiom

The overall figures of DMWEs are sorted on the similar basis of SMWEs classification as depicted below.

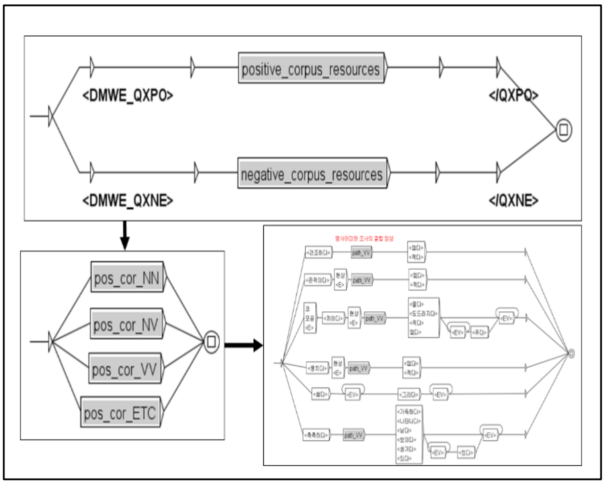


Figure 5: Overall DMWE LGG excerpt

Figure 5 depicts the main LGG with the sub-graphs (e.g. Positive DMWE sub-graphs) analyzing many DMWE sequences. The LGGs can process 1,576 types of DMWEs.

4.2 Compound Named Entity & Feature MWEs

In addition to polarity MWEs, complex named entities and feature nouns should also be correctly analyzed in FBSA. In particular, the majority of cosmetic brands and product names are made up of several words and are right-headed, exhibiting 'Modifier-Head' structure. As opposed to Polarity MWEs that may be noun phrases, verbal phrases or adjectival phrases, these MWEs are basically noun phrases. In addition, they are not particularly related to subjective opinion, but to topics. In FBSA, they mostly play a role of Target (e) or Aspect (a) of the opinion or sentiment. Therefore, it will be crucial to properly recognize named entity MWEs and feature noun MWEs to succeed in FBSA. In this study, we divided these MWEs into 2 sub-categories: Named Entity MWEs (EMWEs) and Feature Noun MWEs (FMWEs).

4.2.1 Named Entity MWEs (EMWEs)

Based on the term-frequency table of EMWEs, we observe that components may be basically brand names, modifiers, heads (referents) or post-modifiers. Generally, a head denotes the referent of the entity such as 'cream', 'toner', 'foundation' or 'mascara' whereas a modifier represents aspects of its referent such as 'moisture' or 'essential'. In Korean, most EMWEs are borrowed from English words, and therefore variations of transliteration weaken recall in automatic recognition of these terms. In addition to these morpho-phonological variations, elision or contraction of certain units occurs in user-generated review texts. These irregularities can be legibly and successfully controlled with the LGG formalism. In the LGG in Figure 6, the combinations of the variable elements are described in a finite-state way. The application of the LGG delimits and normalizes the combinations with the XML-like tags <EMWE-XXPR> and </XXPR>. ('XXPR' annotates a sub-category of Named Entity registered in DECO-Lex: 'Product/Brand Name').

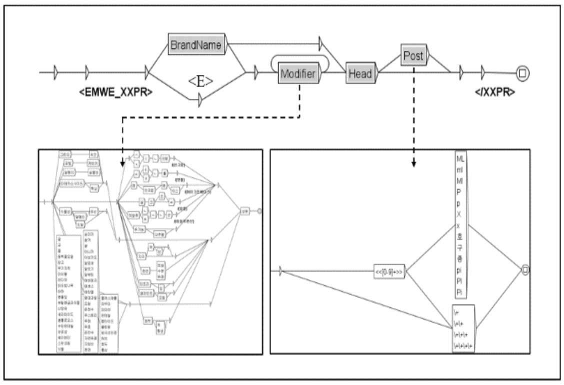


Figure 6: An example of LGGs for EMWE

The LGG in Figure 6 represents the EMWEs consisting of combinations of a brand name, modifiers, a head and a post-modifier. The grey boxes call the sub-graphs. The <E> path makes the brand name optional, which covers EMWEs made up of a product name. The Brand Name LGG is a sub-graph which contains the 468 units of cosmetic brand names observed in the cosmetic review website. Other sub-graphs represent the aggregation of multiple LGGs that recognize 31,560 product names. The LGG in Figure 6 not only includes the lists of products but also retrieves diverse variations caused by elision or contraction of the most frequent tokens. This LGG chunks EMWEs and assigns them a category by the XML-like tags <EMWE_XXPR> and </XXPR>.

The LGG in the left bottom of Figure 6 displays a part of the Modifier LGG: it recognizes modifiers most frequently collocating with a referent. The modifiers denote certain features of the referent, including function and substance. Let us consider:

(4a) 헤라 셀/ *heyla seyl* ('Hera(a brand name) Cell')

(4b) 헤라 셀 에센스/ *heyla seyl eyseynseu* ('Hera Cell (a modifier) Essence')

(4c) 헤라 에센스/ *heyla eyseynseu* ('Hera Essence(a referent)')

(4d) 셀 에센스/ *seyl eyseynseu* ('Cell Essence')

All these examples are legibly treated in the form of LGGs, since the combinatorial properties of each element are directly represented by finite-state transducers. In this way, a great number of complex modifiers may be formalized in LGGs. Predictable complex types may be added in this LGG, even if they are not observed in the corpus.

As a result, EMWEs formalized by LGGs in this study include around 31,560 types.

4.2.2 Feature Noun MWEs (FMWEs)

EMWEs show a relatively small range of phonological variation since many are proper names for which local partners have already chosen a transliteration. In contrast, FMWEs show much more variations since most are English common nouns that users can choose how to transliterate into Korean. Thus, it is outstandingly beneficial to consider their morpho-phonological variations in order to process them properly.

Let us consider the example of ‘color’ that is one of the most frequent feature nouns in this domain. In Korean, this term, as an English loanword, may occur under several forms due to vowel and consonant variations. Consider:

(5a) 컬러감/ *kheolleo-kam* (‘color-feeling=color’)

(5b) 칼라감/ *khalla-kam* (‘color-feeling=color’)

(5c) 칼라 정도/ *khalla-ceongto* (‘color-degree=color’)

(5d) 컬러밝기/ *kheolleo-palkki* (‘color-brightness=color’)

The LGGs representing Feature (a) of product manage to cover the whole case of combinable types including a series of strings in several units of word whether they include white-space or not. These variations can be controlled by the following LGG (Figure 7).

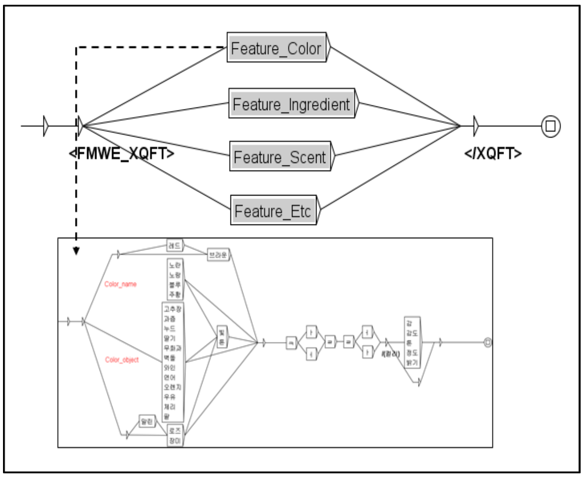


Figure 7: An example of an LGG for FMWE

The sub-graph included in the LGG on Figure 7 is organized in the form of word parts connected together and describes complex feature nouns such as ‘오렌지 칼라/*oleynci khalla* (orange color)’ or ‘블루 컬러감/ *peullwu kheulleu-kam* (blue color-feeling = blue color)’.

We formulate the FMWE composition by frequent collocations of Feature’s headwords, grounded in the term-frequency table.

These FMWEs are chunked and assigned a category with tags which will be crucial to normalizing these variations, such as <FMWE_XQFT> and </XQFT>.

In this study, FMWEs processed by LGGs involve around 165 types.

5. Evaluation

In order to evaluate the linguistic resources proposed in this study, we requested thirty cosmetics reviewers to build a test corpus for the performance evaluation of our resources. The corpus consists of 5,870 tokens(300 sentences) and contains several polarity MWEs and compound noun MWEs as follows:

Polarity MWE		CompoundN MWE		Total
SMWE	DMWE	EMWE	FMWE	427
36	79	266	46	

Table 1: Number of MWEs in the test corpus

Three researchers who majored in linguistics were responsible for the labor-intensive annotation to tag MWE on the test corpus. They cross-checked the tagged corpus based on the strict inter-annotator agreement to be fully served as the evaluation criteria.

We compared the result automatically obtained by the application of LGGs into this corpus with the manually detected result in Table 1. Table 2 shows the result of this evaluation:

	SMWE	DMWE	EMWE	FMWE	Total
Precision	0.933	0.936	0.797	0.948	0.845
Recall	0.777	0.746	0.770	0.804	0.770
F-Measure	0.848	0.830	0.783	0.870	0.806

Table 2: Performance evaluation

The F-measure turns out to be 0.806 while recall is 0.770 and precision 0.845.

As the result, the overall recall shows lower than the precision, but it seems similar to precision (0.797) and recall (0.783) in the case of the EMWEs. Thus, it caused the F-Measure of EMWE to be lower than other types. The main reason for this result is attributed to syntactic ambiguity. In the case of a sentence like ‘왠지 모르게 언니 마스카라가 더 좋더라구요/*waynci molukey enni masukhala-ka te cohtelakuyo* (For some reason, I like a sister mascara)’, the phrase ‘언니 마스카라/*enni masukhala* (sister mascara)’ is ambiguous to be analyzed by two approaches. One way is to parse ‘[$_{np}$ [$_{n}$ 언니] [$_{n}$ 마스카라]]/*enni masukhala* (a sister mascara)’ as an EMWE which refers to a ‘brand’(*enni*) mascara, and the other way is to analyze ′[$_{np}$ [$_{np}$ [$_{n}$ 언니][$_{pos}$ (의)]] [$_{n}$ 마스카라]]/*enni(uy) masukhala-ka* (sister’s mascara)’ caused by ellipsis in noun phrases with possessive case: Korean genitive *josa* ‘의/*uy*’, meaning ‘a sister′s mascara’. Such linguistic ambiguity caused by Korean *josa* omission carries difficulty with recognizing EMWEs for the accurate result.

6. Conclusion

This paper presents a linguistic resource of Korean Multiword Expressions for Feature-Based Sentiment Analysis (FBSA): DECO-MWE. To construct linguistic resources of sentiment MWEs efficiently, we utilized the Local Grammar Graph (LGG) methodology: DECO-MWE is formalized as a Finite-State Transducer that represents lexical-syntactic restrictions on MWEs.

In this study, we built a corpus of cosmetics review texts, which show particularly frequent occurrences of MWEs. Based on the empirical examination of the corpus, four types of MWEs have been discerned. The DECO-MWE thus covers the following four categories: Standard Polarity MWEs (SMWEs), Domain-Dependent Polarity MWEs (DMWEs), Compound Named Entity MWEs

(EMWEs) and Compound Feature MWEs (FMWEs). The retrieval performance of the DECO-MWE shows 0.806 f-measure in the test corpus.

This study brings a two-fold outcome: first, a sizeable general-purpose polarity MWE lexicon, which may be broadly used in FBSA; second, a finite-state methodology adopted in this study to treat domain-dependent MWEs such as idiosyncratic polarity expressions, named entity expressions or feature expressions, and which may be reused in describing linguistic properties of other domains.

7. Bibliographical References

Baldwin, T. and Kim, S. (2010). *Handbook of natural language processing*. CRC Press, Boca Raton, USA, 2nd edition.

Bejček, E. and Pavel, S. (2010). Annotation of multiword expressions in the Prague Dependency Treebank. *Language Resources and Evaluation*, 44(1-2):7–21.

Burnard. L. (2000). *User Reference Guide for the British National Corpus*. Technical report, Oxford University Computing Services.

De Marneffe, M., Pado, S. and Manning, C. (2008). Multiword expressions in textual inference: Much ado about nothing? *In Proceedings of the 2009 Workshop on Applied Textual Inference (TextInfer'09)*, pp. 1-9.

Gross, M. (1997). The Construction of local grammars. *Finite-State language processing*, Roche & Schabes (eds.), the MIT Press.

Gross, M. (1999). Nouvelles applications des graphes d'automates finis à la description linguistique, *Lingvisticae Investigationes* Tome XXII-Vol. Spécial: *Analyse lexicale et syntaxique*: Le système INTEX. Fairon (ed.), John Benjamins Publishing Company, Amsterdam/Philadelphia.

Hu, M. and Liu, B. (2006). Opinion Extraction and Summarization on the Web. *In Proceedings of 21st National Conference on Artificial Intelligence (AAAI'06)*, pages 1621-1624.

Jackendoff, R. (1997). *The Architecture of the language faculty*. Cambridge, USA: MIT Press.

Kim, K. (2017). *An analysis on international competitiveness of Korean cosmetics industry by diamond model*. Graduate college, Sungkyungwan University.

Kim, M., Kim, J., Cha, M. and Chae, S. (2009). An emotion scanning system on text documents. *Korea Emotion and Sensibility*, 12(4):433–442.

Kim, Y. and Shin, H. (2013). Romanization-based approach to morphological analysis in Korean SMS Text, Processing. *In Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pp.145--152

Kim, H. (2000). *A Statistical Study of Idiomatic Phrases in Current Korean*. Graduate college, Yeonsei University.

Lapata, M. and Lascarides, A. (2003). Detecting novel compounds: The role of distributional evidence. *In Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics*, pages 235–242.

Lee, K. (2001). Natural language processing and punctuation marks. *International Association of Language and Literature*, (24):5–38.

Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 5(1):1–167

Liu, B. (2015). *Sentiment analysis: mining opinions, sentiments, and emotions*. Cambridge University Press, NY.

Nam, J. (2012). Study on automatic recognition of Korean negation markers shifting opinion polarity. *Language and Linguistics*, 57, pp. 61--94.

Nam, J. (2015). *Korean electronic dictionary DECO TR-2015-02*. Digital language and knowledge Contents Research Association (DICORA), Hankuk University of Foreign Studies.

Paumier, S. (2003). *De la reconnaissance des formes linguistiques à l'analyse syntaxique*. Ph.D. theses, Université Paris-Est Marne-la-Vallée, France.

Pang, B. Lee, L. and Vaithyanathan, S. (2002). Tumbs Up ?: sentiment classification using machine learning techniques. *In Proceedings of the ACL'02 conference on Empirical methods in natural language processing(EMNLP'02)*, (10):79–86.

Piao, S., Rayson, P., Archer, D., Wilson, A., and McEnery, T. (2003). Extracting multiword expressions with a semantic tagger. *In Proceedings of the ACL'03 Workshop on Multiword Expressions: analysis, acquisition and treatment*, pp. 49—56. Sapporo, Japan.

Sag, A, I., Baldwin, T., Bond, F., Copestake, A. and Flickinger, D. (2002). Multiword expressions: A Pain in the neck for NLP. *Lecture Notes In Computer Science*, (2276):1–15.

Taboada, M., Brooke, J., Tofiloski, M., Voll, K. and Stede, M. (2011). Lexicon-based methods for sentiment analysis. *Computational Linguistics*, 37 (2):267–307.

Tanaka, T. and Baldwin, T. (2003). Noun-Noun Compound Machine Translation: A Feasibility study on shallow processing. *In Proceedings of the ACL'2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment*, pages 17–24, Sapporo, Japan.

Williams, L., Bannister, C., Arribas-Ayllon, M., Preece, A. and Spasić, I. (2015). The role of idioms in sentiment analysis. *In Expert Systems with Applications*, 42(21): 7375–7385.

A Hybrid Approach to Sentiment Analysis Enhanced by Sentiment Lexicons and Polarity Shifting Devices

Gwanghoon Yoo and Jeesun Nam
Graduate School of Linguistics and Cognitive Science, Hankuk University of Foreign Studies, Korea
rhkdgns2008@naver.com, namjs@hufs.ac.kr

Abstract

This paper presents a hybrid approach to sentiment classification method for Korean texts. It is based on a cascading system by which lexicon-based classification first conducts the sentiment detection along with the local parsing of sentiment constituents, and a supervised machine learning algorithm sorts the texts out of the lexicon. We use a fine-grained Korean machine-readable dictionary for the lexicon-based classification, dealing with Polarity Shifting Devices (PSDs) which are divided into Intensifier, Switcher, Activator, and Nullifier. By structuring PSDs and polarity values of opinion texts, it is possible to process complex sentiment constituents efficiently, such as a structure resulting from double negation. Through the performance evaluation, we prove this hybrid approach particularly enhanced by sentiment lexicons and PSDs outperforms the baselines.

Keywords: Sentiment Analysis, Sentiment Lexicon, Polarity-Shifting Device, Hybrid Approach

1. Introduction

This paper aims to propose a novel hybrid approach for Korean sentiment analysis through enhanced Korean sentiment lexicons and Polarity Shifting Devices (PSDs). Based on a fine-grained Korean electronic lexicon DECO that is conceived and constructed on rigorous linguistic criteria (Nam, 2015), this study presents a DECO-PSD classifier which incorporates graphs designing Recursive Transition Network to structure opinion corpora, processes complex sentiment constituents efficiently, and makes use of a supervised machine learning classification for the texts uncovered by the linguistic resource.

Since the advent of Web technologies, an enormous amount of data has been flooding the internet, containing opinions or sentiments of the public. To understand the public sentiment from the data, sentiment analysis research has been flourished. Studies conducted on sentiment analysis in English showed explosive growth up to sixfold in 2014 compared with 2010 (Piryani et al., 2017).

Sentiment analysis focuses mainly on identifying polarity including positive and negative in a document or sentence. It is to detect consumers' feelings and opinions about products or services attributed to typically text-based User Generated Contents. This raises the need to implement automatic tools for identifying the sentiment expressed in text. The classification of a document or a sentence according to its polarity can be conducted by machine learning algorithms, lexicon based methods, or even hybrid methods.

Most of supervised machine learning approaches are based on algorithms such as Naive Bayes, Maximum Entropy, and Support Vector Machine, training on a considerable amount of particular dataset (Hatzivassiloglou and Mckeown, 1997; Pak and Paroubek, 2010; Wang and Summers, 2012). Unsupervised machine learning approaches include algorithms like Pointwise Mutual Information (PMI) which estimates the polarity values of a word by computing the relation to the seed tokens exhibiting the explicit polarity (Turney, 2002). Lexicon-Based methods, on the other hand, depend heavily on linguistic resources including a sentiment lexicon composed of pairs of words and its polarity values. Since particular words exhibit polarity values, it is genuinely essential to construct sentiment lexicon data meticulously. Moreover, lexicon-based methods take into account compositional roles of contextual valence shifting (Polanyi and Zaenen, 2004). For example, negating and intensifying words get involved in contextual sentiment valence shifting of a sentence significantly.

Each of two approaches has advantages and disadvantages. In the case of machine learning-based sentiment analysis, the polarity values of sentiment lexicon are primarily computed through the statistic estimation, which is advantageous in that the coverage can be widened depending on the size of training data, and it minimizes human labor to build a sentiment linguistic resource. However, it has a limitation in dealing with the linguistic compositional rules such as negation and intensification (Neviarouskaya et al., 2015). Plus, when the classifier training on a specific dataset is utilized for another domain, its performance is more likely to drop significantly. On the other hand, lexicon-based methods have the advantage of processing the compositional rules and ensuring transparency of classification criteria. Additionally, they show robust performance across domains and texts (Taboada et al., 2011). However, manual construction of sentiment lexicons requires extensive headwork with relatively limited coverage on informal forms of sentiment words. In this respect, it is necessary to develop a hybrid approach of two methods which complements the disadvantages of each methodology but combines merits.

The hybrid approach includes the machine learning and lexicon-based method containing manually written linguistic rules (Prabowo and Thelwall, 2009). Different sentiment classifiers grounded in lexicon-based or machine learning methods are used in a cascade manner so that when one classifier fails, the next one takes a turn to classify, and so on until the remaining document is categorized.

In this paper, we propose sentiment analysis in a sentence-level, based on the hybrid approach for Korean texts. We make use of DECO-PSD classifier of DecoTex platform[1] (Yoo and Nam, 2017) which processes compositional phenomena by using PSDs and sentiment words registered in DECO-SentLex (Nam, 2015) as well as takes advantage of a supervised machine learning algorithm. It has a cascading system, which is primarily grounded in a lexicon-based classifier utilizing DECO language resource holding lexical information to process compositional rules. For the sentences uncovered by the DECO dictionary-based classifier, a Naïve Bayes classifier gets involved to expand the scalability of polarity classification as a supervised machine learning algorithm training on datasets.

In this paper, Section 2 describes some related studies to this proposal. Section 3 illustrates composition model dealing with valence shifting. Section 4 presents the way to structure opinion texts for DECO PSD classification. Section 5 explains how DECO PSD classification works, and Section 6 presents the results and the comparative evaluation of the different versions of the classifier. Finally, Section 7 concludes this paper and points to some future works.

2. Related Studies

By introducing SO-CAL (Sentiment Orientation CALculator), Taboada et al. (2011) points out that a lexicon-based method is beneficial in processing local context of a sentiment word. This system analyzes the sentiment based on the structured words, which is annotated with their polarity values, incorporating negation and intensification. The system deals with compositional rules of several linguistic contexts that can have an influence on calculating polarity values. SO-CAL shifts the polarity values to the opposite orientation for negation: for example, 'not good' has -3 polarity value due to 'good' with +3 polarity value. Amplifiers like 'so' in English magnify the sentiment intensity whereas downtoners like 'somewhat' decrease it. SO-CAL processes amplifiers and downtoners as a modifier which shifts the sentiment values, and it deals with some words that are unlikely to fit the purpose of sentiment analysis in a sentence, such as modality verbs. The system is programmed to ignore the polarity values for the sentiment lexicon collocated with them. Its performance turned out to be consistent and robust across domains. However, its coverage of sentiment words is restricted in the handcrafted sentiment dictionary, which limits to process various informal forms of sentiment words or coinages.

Moilanen and Pulman (2007) describes a composition model, which computes the polarity values of syntactic constituents from the head polarity of their sub-constituents. The sentiment composition model parses sub-constituents to represent the higher constituent and evaluates the output polarity of the composed constituent. It covers polarity reversal, propagation, and polarity conflict resolution within multiple linguistic constituent types. However, its practicality is bounded by the quality of a syntactic parsing performance. Since most of the text data remain highly unstructured with low grammaticality, complex syntactic parsing seems to be hardly efficient or practical.

Choi and Cardie (2008) deals with compositional rules in the orientation of sentiment expressions by computing the polarity values of the constituents of the expressions and applying inference rules to combine the constituents. The inference rules are specialized in a local syntactic pattern of a sentence. In particular, it points out the vital role of content-word negators, which switch the sentiment orientation of collocating words. For example, when the system detects a pattern like '[eliminate]VP [the doubt]NP' the polarity value can be computed by the inference rule 'Compose([eliminate],[doubt])' which flips the negative value of 'doubt'. The result based on compositional semantics shows better performance than baselines without the consideration of compositional semantics.

While the research above focuses mainly on processing compositional semantics of sentiment expressions limited in a sentiment dictionary, Prabowo and Thelwall (2009) introduces a hybrid or combined approach which makes use of multiple sentiment classifiers including lexicon-based classifiers and machine learning algorithmic classifiers in a sequence of performing best. When one classifier fails to classify a document, it will pass the document onto the next classifier, until the document is sorted. However, it rather focuses on machine learning based classifier than on the quality of lexicon or rule-based classifier. Its sentiment lexicon contains the limited number of sentiment words (3672 entries) and, the rule-based classifier can process a small set of compositional rules not covering content-word negating and flow-flipping by conjunctions like 'but'.

Lu and Tsou (2010) also have an investigation on a hybrid method for sentiment analysis which takes advantage of both the handcrafted sentiment lexicons and annotated corpus to extract sentiments, based on supervised machine learning algorithms. The Chinese sentiment lexicon (31,802 entries) is first adjusted under a machine learning algorithm according to annotated corpora as the training data and then integrated into machine learning models to detect polarity. As a result, the hybrid approach significantly outperforms the baselines. However, it does not take into consideration important compositional rules including intensification or negation.

Dhaoui et al. (2017) empirically evaluates the lexicon-based, machine learning and hybrid approaches using a sample (850 comments) of UGC on Facebook fashion brand pages. It shows that the hybrid approach has significantly improved the performance especially in classifying positive orientation. Its lexicon-based classifier is based on a sentiment linguistic resource of Linguistic

[1] It is available to download from Digital Language and Knowledge Contents Research Association (DICORA) in HUFS. homepage: http://dicora.hufs.ac.kr/

Inquiry and Word Count 2015 (LIWC), which has a limitation to process compositional rules of sentiment constituents.

3. Composition Model

The processing of compositional rules, which deals with valence shifting (Polanyi and Zaenen, 2004), is essential in sentiment analysis. Certain words shift polarity values in a context, called Polarity Shifting Devices (PSD) (Nam, 2012). Neviarouskaya et al. (2015) classifies them into two types: 'Intensifying type' which contains adverbial intensifiers like 'very', 'so' and verbs like 'increase' and 'magnify', and 'Reversing type' which includes grammatical negators such as 'not' and 'no', or content-word negators such as 'eliminate' and 'reduce' in English. They have functional roles in sentiment semantics. On top of two types, we add two more types which shift polarity values in Korean. Consequently, in this study, the types of PSD are divided into four categories: Intensifier, Switcher, Nullifier, and Activator as explained below.

1. Intensifier: PSD which intensifies polarity values, including amplifiers and downtoners
   e.g. 완전/*wanceon* (fully), 매우/*maywu* (so), 조금/*cokeum* (little), 덜/*teol* (less), etc.
2. Switcher: PSD which switches the orientation of polarity, including grammatical negators and lexical (content-word) negators
   e.g. "not": 않다/*anhta*, 아니하다/*anihata*, 못하다/*moshata*, 안/*an*, 아니/*ani*, 못/*mos*, 아니다/*anita*, "there is no": 없다/*eopta*, 제거하다/*ceykeohata* (eliminate), etc.
3. Nullifier: PSD which nullifies polarity values, including imperative, suggestive, and interrogative markers or auxiliary verbs
   e.g. -해야 한다/*-hayya hata* (should), -면/*-myeon* (if), -ㄹ 듯/*-il teus* (seem like) etc.
4. Activator: PSD which activates polarity values out of neutral words
   e.g. 너무/*neomwu* (too) + measuring adjectives, 인생/*insayng* (life) + product nouns

First, Intensifier magnifies or minifies the polarity values of sentiment words in contexts. We take into account Intensifier including amplifiers such as '완전/*wanceon*', '진짜/jinjja', '너무/*neomwu*', etc. and downtoners such as '조금/*cokeum*', '덜/*teol*', etc. When collocating with polarity words, the intensifier-amplifiers add '+1' to the polarity values of nearby sentiment words, and the intensifier- downtoners add '-1'.

Second, Switcher reverses the orientation of polarity values. It includes the function words classified as a negator. Negation can be classified into grammatical negation and lexical negation in Korean. In the case of grammatical negation, there are the 'Short Negation' (e.g. '안 좋다/*an cohta*' meaning 'not good') of adverbial negators such as '안/*an*', '못/*mos*', and '아니/*ani*' as well as 'Long Negation' (e.g. '좋지 않다/*cohci anhta*' meaning 'not good') of negative auxiliary verbs: both of which words negate a predicate (Verb or Adjective). '아니다/*anita*' and '없다/*eopta*' which negate nouns as the complements (e.g. '최고가 아니다/*choyko-ka anita*' meaning 'not the best') are also classified as grammatical negation. In the case of lexical negation, however, a lexical (content-word) negator such as '없애다/*eopsayta*' (get rid of) or '제거하다/*ceykeohata*' (eliminate) reverses the polarity values of sentiment words in a clause (e.g. '고통을 없애다/*kothong-eul eopdayta*' meaning 'get rid of the pain').

Grammatical negation can be relatively simple to formalize due to the restricted number of negators; on the other hand, lexical negation is hard to predict. Nevertheless, DECO dictionary covers a considerable amount of words which function as the content-word negator. In this paper, to save negative values of negators, for the text with no sentiment words but only Switcher included, the lexicon-based classifier of DECO PSD classifier is programmed to assign negative values to the PSD. This makes it possible to process text with negative values without a sentiment word like '말 같지도 않다/*mal kath-cito anhta*' (It does not make sense) or '다신 안 갈 것임/*tasin an kal keos-im*' (I will not visit again).

Third, Nullifier ignores polarity values. It includes functional markers used to make a sentence imperative or interrogative. In Korean, the question mark after sentiment predicate can function as Nullifier. For example, when it comes to a sentence like '그 호텔 좋음?/*keu hotheyl coheum*?' (Is the hotel good?), the positive values of '좋음' is more likely to be ignored since the purpose of the sentence is to ask whether the hotel is a pleasant place or not. Additionally, concessive conjunctions function as flow-flipping devices which nullify the polarity values of preceding sentiment words in the range of a sentence. In Korean, concessive ending suffixes combined with predicates are used as flow-flipping Nullifier like 'but' in English, including '-지만/*-ciman*', '-더라도/*-teolato*', '-ㄴ데/*-ntey*', etc. In a sentence '아름답고 예쁘더라도 싫다/*aleumtap-ko yeppeu-teolato silh-ta*' (I hate it although it looks beautiful and pretty), concessive ending suffix '-더라도/*-teolato*' (although) nullifies all polarity values of preceding words such as '아름답다/*aleumtapta*' (beautiful), '예쁘다/*yeppeuta*' (pretty).

Forth, Activator, which activates the certain orientation of polarity values of neutral words, is divided into Positive Activator and Negative Activator, and the composition of polarized sequences by them is highly predictable. In Korean, '인생/*insayng*' polarizes the following noun related to a product as Positive Activator. The sequence of 인생 and product nouns (e.g. 인생 시계/*insayng sikye*, 인생 치마/*insayng chima*, 인생 화장품/*insayng hwacangphwum*, 인생 영화/*insayng yeonghwa*, 인생 휴대폰/*insayng hyutayphon*, etc.) means 'something of my life' in English, exhibiting explicit positive polarity.

As Negative Activator, on the other hand, '너무/*neomwu*' is a good example. '너무/*neomwu*' mainly functions as an intensifier of sentiment lexicon like 'so' in English;

Content	Code	INT [Intensifier]		SWIT [Switcher]		NULL [Nullifier]		ACT [Activator]	
	Position	F [front]	B [back]	F [front]	B [back]	F [front]	B [back]	F [front]	B [back]
Polarity	**Pol ↑**	UINTF (완전/*wanceon etc.)*	UINTB (늘어나다/*nulenata etc.)*	SWITF (안/*an*, 못/*mos*, 아니/*ani*)	SWITB (않다/*anhda*, 못하다/*moshada*,, 아니다/*anida*, 소멸하다/*somyeolhada* etc.)	NULLF (to nullify a polarity value of preceding words)	NULLB (imperative, interrogative marker, etc.)	PACTF (인생/*insayng* +Product noun)	PACTB (to make a following word positive)
	Pol ↓	DINTF (조금/*cokeum etc.)*	DINTB (줄어들다/*cwuletulta etc.)*			ZABSO (~지만/~ciman, ~ㅓ도/~eodo, etc.)		NACTF (너무/*neomwu* +Measuring adjective)	NACTB (to make a following word negative)

Table 1: Codes for PSD

	Rules			Examples
1	INT(UINTF[Adv],POS[VP])	→	UPOS[Adv,VP]	매우 좋다/*maywu cohta* (so good)
2	INT(DINTF[Adv],POS[VP])	→	DPOS[Adv,VP]	약간 좋다/*yakkan cohta* (somewhat good)
3	SWI(SWITF[Adv],NEG[VP])	→	POS[Adv,VP]	안 나쁘다/*an napputa* (not bad)
4-1	SWI(SWITF[Adv],POS[VP]) UK SWITB[VP]	→	NEG[Adv,VP] UK SWITB[VP]	안 좋은 것이 없다/*an cohun kesi epsta* (there is no a not good thing)
4-2	SWI(NEG[Adv,VP], UK SWITB[VP])	→	POS[Adv,VP,UK,VP]	
5	NUL(NEG[VP],NULLB[AVP])	→	NEU[VP,AVP]	승리해야 한다/*sunglihayya hanta* (should win)
6	ACT(PACT[NP],NEU[NP;PRODUCT])	→	POS[NP,NP]	인생 게임/*insayng keyim* (the game of one's life)

Table 2: Examples of Compositional Rule

however, it also means the degree of excessiveness as an adverb like 'too' when collocating with a measuring adjective (Nam, 2012). For example, '좋아요/*cohayo'* and '나빠요/*nappayo*' are explicit sentiment words implying 'good' and 'bad'. When '너무' collocates them, it just amplifies the polarity values of the sentiment words like '너무 좋아요/*neomwu cohayo'* (so good), '너무 나빠요 /*neomwu nappayo'* (so bad). However, it activates the negative polarity of a measuring adjective: for instance, '길어요/*kileoy"* means 'long', which is hard to be classified as an explicit sentiment word, but it holds negative polarity through being modified by '너무/*neomwu'*. '너무 길어요 /*neomwu kileoyo'* means 'too long', expressing the length of something is excessive.

Korean measurement adjectives like '길다/*kilta'* (long), '짧다/*ccalpta'* (short), '크다/*kheuta'* (big), etc. are basically neutral words, but they possess polarity values when modified by '너무/*neomwu'*. To process the sequence, it is essential to list up measuring adjectives. We use 1384 entries of measuring adjectives registered in DECO dictionary as well as their adverbial forms to process the sequence.

In order to formalize PSD, the processing code for each type is assigned to the corresponding PSD. Through annotating the PDS codes to relevant words, DECO PSD classifier locally parses the sentiment constituents. When collocating with sentiment words, each code shifts polarity values as well as controls the way of polarity shifting by fixing the direction of shifting polarity values of a neighboring sentiment word. It is formalized through the position information code (F/B) attached to the basic PSD type code.

Table 1 describes the category codes of PSD. Words assigned to the four PSD types can be continually updated based on the bootstrap approach, which supports continuous performance improvement. Except for Activator, the other three types of PSD shift polarity values of the collocating sentiment words assigned to corresponding polarity values of DecoPolClass. The intervention of 1 or 2 unassigned tokens (UK) into the combination of PSD and polarity word is allowed. Even though the code ZABSO belongs to the subcategory of Nullifier, it does not depend on the position code, and all polarity values of sentiment words preceding a word assigned to ZABSO are nullified in a range of the sentence boundary. Table 2 shows the samples of compositional rules and their examples. In order to compute the polarity shifting of complex combinations, the PSD processing system iterates five times, thereby parsing the sequences applied by multiple compositional rules such as double negation. For example, in the case of sentence like '안 좋은 것이 없다/*an coh-eun keos-i eopta'* (There is no a not-good thing), the polarity values of '좋은/*coh-eun'* (good) are shifted by double negation but remain same, and PSD DECO classifier process compositional semantics through local parsing as shown the number 4 of Table 2.

4. DECO Annotation using LGGs

For the lexicon-based classifier, we utilize DECO dictionary. It is a Korean Machine Readable Dictionary (MRD), a rich language resource containing various semantic information such as inflection, parts of speech, and syntax data for lexical entries. It also includes semantic categories such as DecoPolClass and DecoPsyClass. We use the sentiment lexicon from the DecoPolClass for sentence-level sentiment analysis. The polarity categories are divided into seven types (‘Strongly-Positive’, ‘Positive’, ‘Strongly-Negative’, ‘Negative’, ‘Neutral’, ‘Dependent Polarity’ and ‘Strongly-Dependent Polarity’). In this paper, four categories play a critical role in the polarity classification: Strongly-Positive (QXSP), Positive (QXPO), Strongly-Negative (QXSN), and Negative (QXNG), which have total 12,999 lexical entries. The following table shows the distribution of the sentiment entries by parts of speech.

DecoPolClass	**Noun**	**Verb**	**Adjective**	**Adverb**	**Total**
Strongly- Positive <QXSP>	174	312	326	550	1362
Positive <QXPO>	1133	1882	1263	1709	5987
Strongly- Negative <QXSN>	254	1040	763	1052	3109
Negative <QXNG>	2887	3531	1649	2097	2541

Table 3: Lexical entries in four polarity categories by the part-of-speech in the DECO dictionary

The DECO dictionary was implemented in a compatible manner with the natural language processing platform, Unitex (Paumier, 2003). Based on DECO dictionary, Unitex performs the morphological analysis in the input text. Its automaton processing Korean alphabets analyzes surface forms of the input text based on the lexical information of DECO dictionary, handling the complex morphological inflection in Korean.

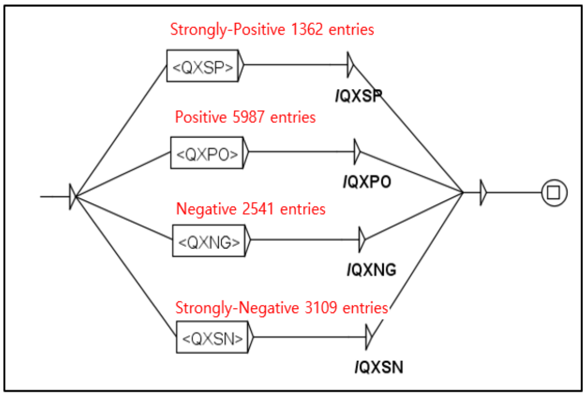


Figure 1: LGG for DecoPolClass Annotation

With the output corpus resulting from morphological analysis, Local-Grammar Graph (LGG) (Gross, 1997, 1999) can be used to extract or adjust DECO lexical information. LGG is a Recursive Transition Network (RTN), converted into Finite-State Automata (FSA) and Finite-State Transducer (FST) to formulize lexical patterns and modify texts (Gross 1997, Nam 2013). Through LGG, a user can generate the marked-up corpus of which tokens are annotated with specific lexical information as DECO codes, referring to as DECO annotation. Figure 1 shows LGG which processes to annotate 12,999 entries of sentiment words with corresponding polarity category codes. The LGG has a total of four paths set to output a specific tag after each path. For example, in the case of the first path, if a strongly positive word allocated to <QXSP> occurs in the input text, the path recognizes the string and attaches ‘/QXSP’ to it.

(1) 재미있지 않다고 하지만 완전 인생 영화였다.
caymiissci anhtako haciman wanceon insayng yeonghwayeossta.
(Not funny, but it was really the movie of my life.)

For example, when processing a sentence (1) as an input value, the morphemes of each token is tokenized and analyzed to result in a structured text (2) through the DECO lexicon which assigns various lexical information to them, including morphological, semantic, and syntactic information.

(2) 재미있/재미있다/AS/ZAZ/LEO/REP/YAEP/YA
PZ/QXPO/QXJO/QPSI/QPPS+지/지/EA/LI/CNS 않/않
다/VS/ZVZ/LEO/REP/YVLZ/QXDE/QX
EL/QXND/QINA+다/다/EV/MI/DDA+고/고
/EV/LI/AND 하/하다/AS/ZAP/LEO/HAP/YACZ+지만/
지만/EA/LI/CNS 완전/완전
/DS/ZDZ/LEO/REP/QXAD/QXEL/QDEG
인생/인생/NS/ZNZ/LEO/SLB/NAB/QBIC/XXCO
/XXCR/XQRL 영화/영화/NS/ZNZ/LEO/SLB/MCO
/NAB/QXDE/QXEL/QCRR/ QART/XQRL+이/이다
/EA/CPA/IDA+ㅓㅆ/었/EA/MT/PAS+다/다/EA/TE
/DEC+./../SB/PUN/DEC{S}

Since it contains too much lexical information, it is efficient to structure the sentence with LGG so that necessary information is extracted for sentiment analysis. LGGs as shown Figure 2 are in a form of RTN to function as FSA and FST which has transitions from the initial state to the final state. They process some of PSD including Intensifier, Switcher, Activator, and Nullifier as explained in Section 3. When the LGGs are merged in the main graph to process the sentence (2), the following output is obtained.

(3) 재미있/QXPO지 않/SWITB다고 하지만/ZABSO 완전/UINTF 인생/PACTF 영화/QXZE였다.

In this way, corpus modification is performed to mark up the lexical information necessary for the input corpus by using the category codes to which the sentiment words and function words of the DECO dictionary are allocated. It is used as the structured text, which is input data of DECO PSD classifier computing polarity values as well as processing linguistic compositional rules of the polarity constituents in the document. Through DECO annotation based on LGG, the four types of PSD are structured for local semantic parsing as (4).

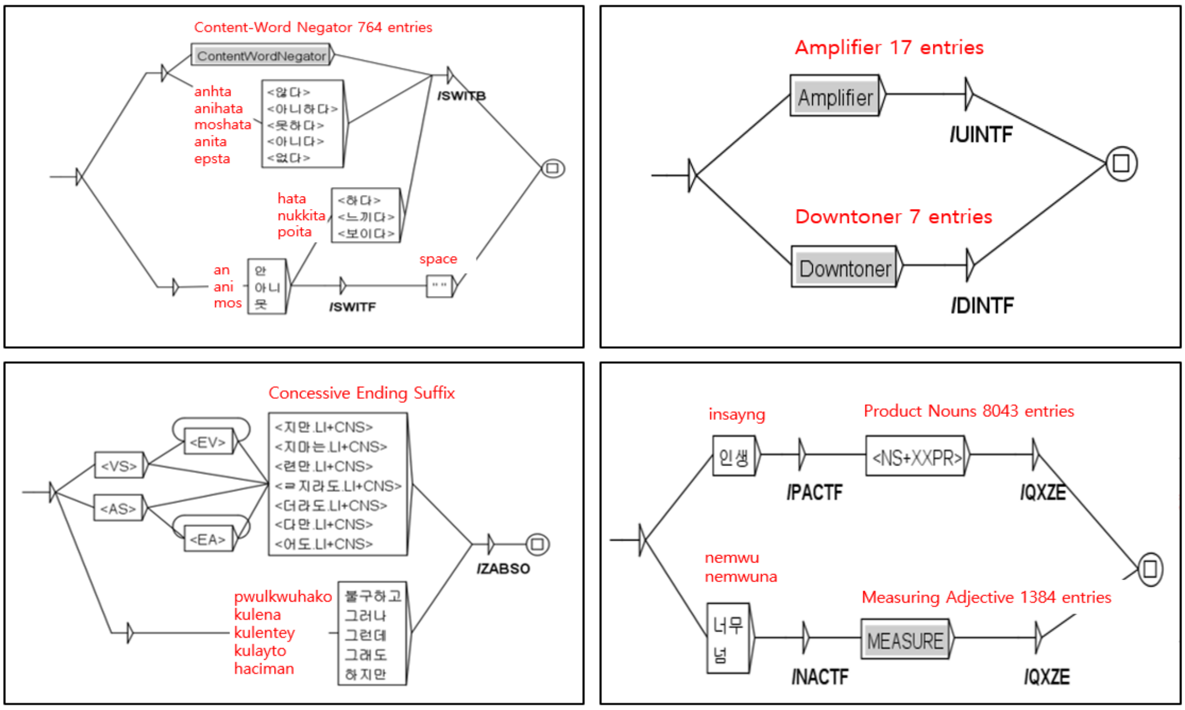


Figure 2: Examples of PSD LGG

(4) ((((재미있/QXPO지: +2) 않/SWITB다고: -2) 하지만 /ZABSO: 0) (완전/UINTF (인생/PACTF 영화/QXZE였다: +2): +3): +3)

For better understanding, even though it is not parallel to (4), the parsing mechanism is somewhat similar to the way in English as following (5).

(5) (((Not (funny: +2): -2), but: 0) it was (really (the movie of my life: +2): +3): +3).

5. Hybrid Sentiment Analysis Model

DECO PSD classifier parses the local sentiment constituents of the structured texts by DECO annotation and vectorizes the polarity values. By aggregation of the values, it classifies the polarity of each sentence. DecoPolClass 'QXSP' (Strongly-Positive), 'QXPO' (Positive), 'QXNG' (Negative), and 'QXSN' (Strongly-Negative) are assigned to polarity values as (1) below.

(1) QXSP: +4, QXPO: +2, QXNG: -2, QXSN: -4

Each polarity orientation has five degrees. Intensifier can result in even values such as +1, +3, +5 or -1, -3, -5. Including value zero, a sentiment word can be assigned from +5 to -5.

Domain	Annotated Text	Result
RES	분위기도 나쁘/QXNG지 않/SWITB았고요. *pwunwikito nappu*/QXNG*ci anh*/SWITB*asskoyo.* (The atmosphere was not bad.)	+2
IPT	단단하다는 느낌은 좋았지만/ZABSO 너무/UINTF 부담스러울/QXNG 정도로 무게감이 느껴져요. *tantanhatanun nukkimun cohassciman*/ZABSO *nemwu*/UINTF *pwutamsulewul*/QXNG *cengtolo mwukeykami nukkyecyeyo.* (The feeling of its solidity was good but so heavy that I even feel it burdensome.)	-3
MOV	보는 동안 지루하/QXNG지 않/SWITB았음 *ponun tongan cilwuha*/QXNG*ci anh*/SWITB*assum* (I was not bored while watching it.)	+2
TRV	시설도 너무/UINTF 좋/QXPO았어요. *siselto nemwu*/UINTF *coh*/QXPO*asseyo.* (The facility was very good.)	+3
IPT	내장 메모리가 너무/NACTF 작/QXZE음 *naycang meymolika nemwu*/NACTF *cak*/QXZE*um* (Internal memory is too small.)	-2
CLO	인생/PACTF 샷/QXZE 찰칵 *insayng*/PACTF *syas*/QXZE *chalkhak* (The snapshot of my life, click.)	+2
TRA	사장님이 착하세요/QXPO ?/NULLB *sacangnimi chakhaseyyo*/QXPO ?/NULLB (Is the host kind?)	0
Domain	**Out Of Dictionary Text**	**Result**
RES	주말에 들리는데 여기 한번 가봐야겠네요. *cwumaley tullinuntey yeki hanpen kapwayakeyssneyyo*. (Stopping by around on the weekend, I should visit there.)	0
MOV	시간 가는 줄 모르고 본 영화. *sikan kanun cwul moluko pon yenghwa*. (The movie I got carried away watching)	0
IPT	저거 키보드 하나 살돈으로 샤오미 패드를 사겠수다. *ceke khipotu hana saltonulo syaomi phaytulul sakeyssswuta.* (With the money to buy the keyboard, I will rather buy a Xiaomi pad.)	0
IPT	사운드바 교환식은 아이디어인 듯. *sawuntupa kyohwansikun aitiein tus.* (Soundbar exchange seems like an idea.)	0
TRA	올라가는데 진짜 땀을 한 바가지 쏟아냈다. *ollakanuntey cincca ttamul han pakaci ssotanay-ssta.* (I sweat a lot during climbing.)	0
CLO	입으면 심지어 허리 쪽 살 울퉁불퉁 잡힘. *ipumyen simcie heli ccok sal wulthwungpwulthwung caphim.* (When wearing it, I can even grab bumpy fat in my waist.)	0

Table 4: Examples of Annotated Text and Text Out of Dictionary

Table 4 shows the examples of annotated sentences as well as those uncovered by DECO annotation. Unlike the sentences assigned to the polarity values, the lexicon-based polarity classification cannot compute a polarity value of the Out Of Dictionary (OOD) sentences. Many of them are attributed to spelling or spacing errors, but some OOD sentences are due to the limitation in processing idiomatic or figurative expressions including '가봐야겠다 /*kapwayakeyssta'* (should visit), '시간 가는 줄 모르다 /*sikan kaneun cwul moleuta'* (get carried away), '아이디어 다/*aitieo-ta'* (it is an idea), '땀을 한 바가지 쏟아냈다 /*ttam-eul han pakaci ssotanay-ssta'* (sweat a lot). In terms of precise sentiment analysis, constructing a linguistic resource to process the multiword expressions is much preferable; however, since it requires a lot of time and human energy, machine learning algorithms can be replaced of it. DECO PSD classifier makes use of a supervised machine learning algorithm - Naïve Bayes to classify the OOD texts.

$$C_{NB} = \underset{\mathrm{c} \in \mathrm{C}}{argmax}P(c)\prod_{\mathrm{f} \in \mathrm{F}}P(w|c) \qquad (1)$$

Naïve Bayes is a well-known algorithm to perform robustly even with a relatively small amount of training data compared with other algorithms such as SVM or Maximum Entropy (Pak and Paroubek, 2010; Wang and Manning, 2012). It is probabilistic classification based on 'Bag of Words' approach. Under Naïve Bayes assumption implying that the tokens in a document are independent of the document class, it can be formulized by Equation 1. In this paper, 18,297 sentences of five domains from *MUSE* (Multilingual Sentiment Lexica & Sentiment-Annotated Corpora) opinion corpus (http://dicora.hufs.ac.kr) sets are used as the training data.

6. Experiment

6.1 Corpora

MUSE Domain	Test		Training	
	Tokens	Sentences	Tokens	Sentences
Restaurant (RES)	11441	1584	28326	3661
IT products (ITP)	11757	1574	27548	3665
Travel (TRA)	7877	942	19095	2210
Clothes (CLO)	15201	1722	35407	4108
Movie (MOV)	11441	1994	27024	4653
Total	57717	7816	137400	18297

Table 4: Testing and training corpus information by each domain

For the performance evaluation of DECO PSD classification, we use the five-domain opinion corpora of MUSE project conducted by DICORA Research Center, which consists of web-scraped reviews and Social Media comments from various websites of various domains. MUSE opinion corpora are manually annotated with the sentiment classification in the sentence-level. To evaluate the robust performance of multi-domain documents, we make use of the comments about restaurants (RES), IT-related products (ITP), travel-related services (TRA), clothes (CLO). 70% of each domain corpus is used as the training data, and the rest of corpus is for testing data as shown Table 4.

6.2 Results

To measure the performance efficiently, we adopt precision, recall, f-measure, and accuracy. Precision is the fraction of correct instances of a polarity among the classified cases of the polarity, whereas recall is the fraction of correctly classified instances of a polarity over the total correct instances of the polarity. F-measure is the harmonic mean of precision and recall, and accuracy is the fraction of the total of correctly classified opinions over the total opinions submitted to the classifier.

Domain	Accuracy		
	NB	NB+DECO	NB+PSD
RES	0.805	0.809	**0.838**
ITP	0.693	0.752	**0.78**
TRA	0.764	**0.818**	0.807
CLO	**0.845**	0.808	0.829
MOV	0.756	0.763	**0.769**

Table 5: Accuracy of five domains

Classifier	Polarity	Recall	Precision	F-measure
NB	Positive	0.795	0.908	0.848
	Negative	0.695	0.472	0.562
	Accuracy	0.774		
NB+DECO	Positive	0.857	0.919	0.887
	Negative	0.772	0.643	0.702
	Accuracy	0.787		
NB+PSD	Positive	0.871	0.907	**0.889**
	Negative	0.763	0.689	**0.724**
	Accuracy	**0.803**		

Table 6: Overall performance evaluation

Table 5 shows the accuracy of each domain, and Table 6 presents the overall performance of whole domain. NB indicates the Naïve Bayes classifier as a baseline classifier, and NB+DECO refers to a combined classifier without processing PSD. Notably, NB+DECO outweighs a baseline classifier even if it cannot deal with PSD processing. As expected, PSD classifier (NB+PSD) shows the best performance over others, which means the hybrid sentiment classification regarding PSD processing yields the robust performance over various domains.

7. Conclusion

This paper proposes the novel approach, hybrid sentiment classification based on DECO PSD classifier processing Polarity Shifting Devices, outperforming baselines. Based on DECO dictionary and Naïve Bayes classification, it has a cascading system through which a lexicon-based classifier locally parses the sentiment constituents to detect opinions first, and then Naïve Bayes classifier sorts the Out Of Dictionary texts by training on MUSE opinion corpora.

In particular, this paper introduces the efficient composition model and how to process it, dealing with four types of PSD including Intensifier, Switcher, Activator, and Nullifier. With simple but powerful compositional rules, it is possible to compute polarity values of complex sentiment constituents such as ‘double negation’.

For future works, it is in high demand to have an in-depth investigation on the lexical items which would be assigned to PSD. Since this paper describes a few examples of them, it is essential to study the various aspects of PSD and expand its lexicon. Additionally, more research should get attention to construct a linguistic resource covering a vast amount of multiword expressions so that the coverage of lexical information can expand to detect the hidden polarity values.

8. Bibliographical References

Choi,Y. and Cardie, C. (2008). Learning with compositional semantics as structural inference for subsentential sentiment analysis. *In Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 793--801.

Dhaoui, C., Webster, C. and Tan, L. (2017). Social media sentiment analysis: lexicon versus machine learning. *Journal of Consumer Marketing*, 34(6):480–488.

Gross, M. (1997). The Construction of local grammars, in Finite-State language processing, *Roche & Schabes (eds.)*, the MIT Press.

Gross, M. (1999). Nouvelles applications des graphes d'automates finis àla description linguistique, Lingvisticae Investigationes Tome XXII-Vol. *Spécial: Analyse lexicale et syntaxique: Le système INTEX, Fairon (ed.)*, John Benjamins Publishing Company, Philadelphia.

Hatzivassiloglou, V. and Mckeown, R. K. (1997). Predicating the semantic orientation of adjectives. In Philip R. Cohen, Wolfgang Wahlster (Program Chair), *In Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics (EACL '97)*, pages 174-181.

Lu, B. and Tsou, B.K. (2010). Combining a large sentiment lexicon and machine learning for subjectivity classification. *In Proceedings of the 9th International Conference on Machine Learning and Cybernetics*, pages 3311–3316.

Moilanen, K. and Pulman, S. (2007). Sentiment composition. *In Proceedings of Recent Advances in Natural Language Processing (RANLP 2007).* September 27-29, Borovets, Bulgaria. pp. 378--382.

Nam, J. (2012). Study on automatic recognition of Korean negation markers shifting opinion polarity. *Language and Linguistics*, 57, pp. 61-94.

Nam, J. (2013). Understanding Maurice Gross' Language Processing Model and its Computational Applications. *Lingua Humanitatis*, 15(1):125—151.

Nam, J (2015). *DECO Korean Electronic Dictionary*. DICORA Technical Report, TR-2015-02. HUFS, Korea.

Neviarouskaya, A., Prendinger, H., and Ishizuka, M. (2015). Attitude Sensing in Text Based on A Compositional Linguistic Approach. *Computational Intelligence*, 31(2):256–300.

Pak, A. and Paroubek, P. (2010). Twitter as a corpus for sentiment analysis and opinion mining. *In Proceedings of the International Conference on Language Resource and Evaluation(LREC'10)*, pages. 1320-1326, Valletta, Malta, may.

Paumier, S. (2003). *De la reconnaissance des formes linguistiques à l'analyse syntaxique*. Ph.D. thesis, Université Paris-Est Marne-la-Vallée, France.

Piryani, R., Madhavi, D. and Singh, V. K. (2017). Analytical mapping of opinion mining and sentiment analysis research during 2000–2015. *Information Processing and Management*, 53, pp. 122-150.

Polanyi, L., and Zaenen, A. (2004). Contextual valence shifters. *In Proceedings of AAAI Spring Symposium on Exploring Attitude and Affect in Text*, pages 106–111.

Prabowo, R. and Thelwall, M. (2009). Sentiment analysis: A combined approach. *Journal of Informetrics*, 3(2):143–157.

Taboada, M., Brooke, J., Tofiloski, M., Voll, K. and Stede, M. (2011). Lexicon-based methods for sentiment analysis. *Computational Linguistics*, 37 (2):267–307.

Turney, P. D. (2002). Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. *In Proceedings of the 40th annual meeting of the Association for Computational Linguistics (ACL'02)*, page 417–424.

Wang, S and Manning, C.D. Baselines and bigrams: Simple, good sentiment and text classification. *In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL '12)*, pages 90-94.

Wang, S. and Summers, M. R. (2012). Machine learning and radiology. *Medical Image Analysis*, 16(5):933 – 951

Yoo, G. and Nam, J. (2017). *DecoTex Users' Manual*, TR-2017-12, Digital Language And Knowledge Contents Research Association (DICORA), Hankuk University of Foreign Studies.

An Easier and Efficient Framework to Annotate Semantic Roles: Evidence from the Chinese AMR Corpus

Li Song[1], Yuan Wen[1], Sijia Ge[1], Bin Li[1], Junsheng Zhou[2], Weiguang Qu[2, 3], Nianwen Xue[4]

1. School of Chinese Language and Literature, Nanjing Normal University, Nanjing, 210024, China

2. School of Computer Science and Technology, Nanjing Normal University, Nanjing, 210023, China

3. Key Laboratory of Information Processing and Intelligent Control, Minjiang University, Fuzhou, 350108, China

4. Computer Science Department, Brandeis University, Waltham, 02453, USA

songli.njnu@gmail.com

Abstract

Semantic role labeling (SRL) is one of fundamental tasks in Chinese language processing. At present, it has three major problems on the construction of the SRL corpus. First, there are disagreements over the definition of the number and frame of semantic roles. Second, static predicate frames are hard to cover dynamic predicate usages. Third, it is unable to annotate the dropped semantic roles. The newly designed Abstract Meaning Representation (AMR) is a novel method of representing the meaning of sentences, which offers dynamic mechanisms to provide better solutions to the above three problems. We use the Chinese AMR corpus of 5,000 sentences to make a detailed comparison between AMR and other SRL resources. Data analysis shows that in AMR, it is easier to annotate the semantic roles of a predicate with the simplified distinction between core roles and non-core roles. And 1,045 tokens of dropped roles are annotated under this new framework. It indicates that AMR offers a better solution for Chinese SRL and sentence meaning processing.

Keywords: Abstract Meaning Representation, predicate framework, semantic role, language knowledgebase

1 Introduction

Automatic semantic analysis is one of the core tasks in Natural Language Processing (NLP). Therefore, building the semantic resources is the first step for machine learning based NLP systems. In semantic representation, semantic relations between predicates and their semantic roles form the backbone of the sentence structure. Thus, building the predicate frames which describe such information becomes an important issue in linguistics and NLP. There have been many semantic role labeling (SRL) systems and SRL resources in different languages, but there are several problems in these SRL corpus.

First, the number of the semantic role labels of predicates is still to be discussed in linguistics. VerbNet uses 30 general thematic role labels to represent semantic relations (Kipper et al., 2000). Sinica Treebank distinguishes necessary and unnecessary arguments and uses 60 semantic role labels, 12 of which can represent necessary arguments (Chen et al. 2003). FrameNet defines semantic roles on a per-frame basis (Baker et al., 1998), so it avoids determining how many semantic roles are needed for a language, and there are 1224 frames in FrameNet and 323 frames in Chinese FrameNet (CFN). PropBank (Palmer et al., 2005) and Chinese Proposition Bank (CPB) (Xue & Palmer, 2009) both define 5 predicate-specific semantic roles for the core arguments and 13 semantic roles that are consistent across predicates for non-core arguments. It can be seen that the number of role labels used by different SRL resources is quite different. This is mainly because these resources are based on different theoretical backgrounds.

Second, it is hard for static predicate frames to cover dynamic predicate usages. Predicate frames which do not distinguish core and non-core roles are difficult to represent whether a semantic role is necessary for the predicate. And resources that define core roles in a predicate-independent manner just as non-core roles neither could solve the collision between core and non-core roles nor could represent multi-functional semantic roles.

Third, limited to the annotating mechanism, most SRL systems are unable to annotate the dropped semantic roles of the predicates. For example, it is hard for most SRL systems to represent correctly the meaning of the nominal phrase *the injured* whose central words are dropped and *one of which…* which drops the noun that appeared in the preceding clause.

Abstract Meaning Representation (AMR), a new method to represent meaning of sentences, defines semantic roles in a manner different from other SRL systems (Banarescu et al., 2013). It deals with core and non-core roles in different specialized ways. AMR annotates core arguments using the same five core role labels as in PropBank, which are predicate-specific, and adopts the predicate frame lexicon extracted from PropBank. But the number of non-core role labels that are general to all the predicates is up to 40. At the same time, AMR allows to add back dropped semantic roles in the sentences. Through the dynamic mechanisms, AMR can provide better solutions to the above three problems. The English AMR Sembank[1] has included 39,260 sentences and become an important semantic resource.

Referring to the guidelines of English AMR, Li et al. (2016) has developed annotation specifications for Chinese AMR (CAMR), taking linguistic characteristics of the Chinese language into account. CAMR uses the same 5 core role labels (arg0-arg4) and 44 non-core role labels (time, location, cause, etc., four of which are added based on the needs of Chinese annotation) as AMR. The predicate frame lexicon of CAMR is extracted from the corpus (Bai & Xue, 2016) of Chinese Proposition Bank (CPB) (Xue & Palmer, 2009). In addition, Li et al. (2017) designs a framework for aligning the concepts and relations to word

[1] https://catalog.ldc.upenn.edu/LDC2017T10

tokens in a sentence for CAMR, which is helpful for annotating dropped semantic roles. Since English AMR can provide better solutions to the above three problems, we try to discuss whether CAMR can provide better solutions to these problems in Chinese.

The rest of this paper is organized as follows. In Section 2, we discuss the related work. In Section 3, we introduce the core and non-core role labels of CAMR and the basic information of the CAMR corpus. In Section 4 and Section 5, we discuss the rationality of the core and non-core role labels of CAMR based on data analysis. Section 6 discusses the advantages of the permission of adding back dropped roles of AMR. The conclusions and future work can be found in Section 7.

2 Related Work

Constructing a predicate frame lexicon combining with labeling semantic roles of predicates in corpus has become a research paradigm. There are many methods to define semantic roles, but the granularity of the semantic roles of predicates are still disputed in the linguistics field. Xue (2006) argues that the specific semantic roles in different SRL resources range from very general role labels to labels that are meaningful to a specific situation to predicate-specific labels in terms of levels of abstraction.

VerbNet uses 30 general thematic role labels such as agent, theme and beneficiary to represent semantic relations (Kipper et al., 2000). Similarly, Sinica Treebank which is a semantic treebank in traditional Chinese defines 60 semantic role labels in a predicate-independent manner. Additionally, Sinica Treebank distinguishes necessary and unnecessary arguments, and uses 12 of the 60 labels to represent necessary arguments (Chen et al. 2003). There are also similar resources in simplified Chinese such as NetBank, which defines 8 kernel thematic roles (agent, patient, recipient, etc.) and 18 circumstantial thematic roles (time, location, reason, etc.), all of which are general for predicates (Yuan, 2007).

FrameNet defines semantic roles on a per-frame basis, so it avoids determining how many semantic roles are needed for a language, leading to a large quantity of semantic role labels. These labels are extracted from specific predicates and applied to the same category of verbs and nouns which have arguments. Chinese FrameNet (CFN) follows the system of FrameNet. There are 1,224 frames in FrameNet and 323 frames in CFN.

PropBank defines semantic roles for the core arguments in a predicate-specific manner. Each sense of each verb has a specific set of roles, which are given only numbers (0-5) rather than names: Arg0-Arg4. Bai & Xue (2016) argues that Core arguments have three main attributes: (1) obligate, meaning of a predicate will be incomplete if it lacks a core argument; (2) different, the core argument frames of predicates differ from one another, so each sense of each predicate has a specific set of roles; (3) exclusive, multiple core arguments do not serve as the same semantic role. Different from core roles, its semantic roles for non-core arguments are consistent across predicates, and there are 13 non-core role labels (ADV, TMP, LOC, etc.) adopted by PropBank. Following the system of PropBank, Chinese Propsition Bank (CPB) adopts the same 5 core roles and 13 non-core roles.

AMR is a novel method of meaning representation which deals with core and non-core roles in different specialized ways. It annotates core arguments using the same five core role labels as in PropBank, which are predicate-specific, and adopts the predicate frame lexicon extracted from PropBank. But the number of non-core role labels (time, location, cause, etc.) which are general to all the predicates is up to 40. Chinese Abstract Meaning Representation (CAMR) uses the same 5 core role labels (arg0-arg4) and 44 non-core role labels, four of which are added for the needs of Chinese AMR annotation.

It can be seen that there are many SRL resources in Chinese as well as English, but their granularity of semantic role labels differs from each other. Table 1 summarizes the main SRL resources in English and Chinese that differ in the granularity of semantic role labels.

Resources	Language	Role Labels
VerbNet	English	30 general role labels
Sinica Treebank	Traditional Chinese	60 general role labels (5 for nouns, 12 for core roles, 43 for non-core roles)
NetBank	Simplified Chinese	8 general core labels and 18 general non-core labels
FrameNet	English	1,224 frames (role labels are frame-specific)
CFN	Chinese	323 frames (role labels are frame-specific)
PropBank	English	5 predicate-specific core labels and 13 general non-core labels
CPB	Chinese	5 predicate-specific core labels and 13 general non-core labels
AMR	English	5 predicate-specific core labels and 40 general non-core labels
CAMR	Chinese	5 predicate-specific core labels and 44 general non-core labels

Table 1: Main SRL Resources in English and Chinese

3 Chinese AMR

3.1 Core and Non-core Roles of Chinese AMR

Following the annotation scheme of OntoNotes adopted by English AMR, CAMR uses predicate senses and core argument frames in CPB, and annotates semantic relations with core and non-core semantic relation labels. Core semantic relations refer to the inevitable semantic relations in the event framework of the predicates which are predicate-specific. Table 2 shows the 5 core semantic relations adopted from CPB. Non-core semantic relations refer to the semantic relations outside the core semantic relations, which are predicate-independent. English AMR defines 40 general non-core semantic relations so that they are fine-grained, and CAMR adds 4 non-core relations taking the characteristics of Chinese into account. In order to be compatible with AMR, CAMR still uses English words to represent labels of non-core semantic relations. Table 3 shows non-core semantic relations in CAMR.

arg0	external argument (Proto-Agent)
arg1	internal argument (Proto-Patient)
arg2	indirect object / beneficiary / instrument / attribute / end state
arg3	start point / beneficiary / instrument / attribute
arg4	end point

Table 2: Core Semantic Relations in CAMR

accompanier	direction	mod	quant
*aspect	domain	mode	range
beneficiary	duration	name	source
cause	example	ord	subevent
compared-to	extent	part-of	subset
consist-of	frequency	path	superset
condition	instrument	*perspective	*tense
cost	li	polarity	time
*cunit	location	polite	topic
degree	manner	poss	unit
destination	medium	purpose	value

* are the added relations in CAMR

Table 3: Non-core Semantic Relations in CAMR

Since core semantic roles are defined with respect to an individual verb sense, AMR and CAMR need support of predicate frame lexicons. The frame lexicon of CAMR is extracted from the CPB corpus, consisting of 26,650 senses of 24,510 predicates.

3.2 The Chinese AMR Corpus

According to the CAMR annotation specifications developed by Li et al. (2016), we extracted 5,088 Chinese sentences from Penn Chinese TreeBank (CTB) 8.0[2] and annotated them. The inter-agreement smatch score of 500 randomly selected sentences between the two annotators is 0.83. The sentences we annotated in CTB are from microblog, which cover a wide range of fields and rich topics. Most sentences are long and complicated, containing rich semantic information. Before annotating, we deleted wrong sentences artificially, and then carried on automatic word segmentation and artificial proofreading. The final corpus consists of 5,000 Chinese sentences. Table 4 shows the basic data of these sentences. Compared with the Chinese version of *the Little Prince* AMR corpus (Li et al., 2017), whose average sentence length is 12.90 words and average number of concepts is 9.48, sentences in this corpus are longer and more complex.

Sentences	5,000	Characters (AVG)	34.34
Characters	171,703	Words (AVG)	22.46
Words	112,348	Concepts (AVG)	18.36
Concepts	91,808	Added Concepts[3] (AVG)	3.02

Table 4: Basic Data of the CAMR Corpus

4 Core Roles in Chinese AMR Cover Dynamic Problems

The definition of core arguments in PropBank has been controversial in linguistics field. Some scholars consider it too broad and not conducive to classification of semantic roles, the predicate frame of AMR thus failed to be approved by the entire linguistics field. Therefore, we try to explore whether the predicate framework adopted by AMR can represent core semantic roles of predicates more reasonably.

We consider that there are two inescapable problems in predicate frameworks whose core role labels are consistent across predicates: (1) the core semantic role labels are applicable for all predicates, and the core roles and non-core roles may conflict when annotating concepts of location, cause, instrument and so on, for example, a concept of location is indispensable to the meaning of *appear*. (2) It is difficult to properly annotate the multi-functional roles, for example, a concept of agent or cause can both serve as the subject of *change*.

These problems are common in Chinese and they can be solved by the predicate framework of CAMR, whose predicate-specific frame lexicon is extracted from the CPB corpus, which contains 26,650 senses of 24,510 Chinese predicates (verbs, adjective, etc.). CPB is a corpus which adds semantic roles of predicates to CTB (Xue et al., 2005), a syntactically annotated Chinese corpus that is word-segmented, POS-tagged and syntactically bracketed with phrase structures (Xue & Palmer, 2009). Here we elaborate how CAMR solves the collision between core and non-core roles and how it annotate multi-functional roles based on statistical data of the predicate frame lexicon and CAMR corpus.

4.1 Solve the Collision between Core and Non-core Roles

Each sense of each predicate in the predicate framework of CAMR has a specific set of roles. If a concept is essential for the meaning of the predicate, it serves as the core role of the predicate, even though it represents the location or cause of the predicate, which is a kind of collision between core and non-core roles. If inessential, it serves as a non-core role of the predicate. For example, the concept of location is indispensable in the meaning of 遍布-01 (the first sense of 遍布, be spread throughout somewhere), so it is a core role of 遍布-01:

遍布-01 (be spread throughout somewhere)
arg0: theme
arg1: **location**

It's even possible that 4 of the 5 roles of a predicate are in conflict, such as 引进-01 (introduce something from one place to another):

引进-01 (introduce something from one place to another)
arg0: agent / **cause**
arg1: entity imported
arg2: **location** arg1 is imported from
arg3: predicate, **purpose**
arg4: **destination**

There are many predicates whose core and non-core roles are conflicting. We count how many predicates in the CPB lexicon have collision between their core and non-core roles. Data shows that the total number of these senses is 2,453, accounting for 9.20% of all the senses in the

[2] http://amr.isi.edu/download.html

[3] There are three main kinds of added concepts in CAMR: (1) added semantic roles, (2) types of named entities which are used to identify the names of an entity, like *country* for *China*, (3) discourse relations such as *condition*, *temporal*.

lexicon of CAMR. Among them, 5.99% have collision between more than two core roles and non-core roles. Additionally, through analyzing all the description of core roles in the CPB lexicon, we find that there are 24 kinds of non-core roles may conflict with the core roles, which means that more than half of the categories of non-core roles are able to enter the core argument frame of predicates. Table 5 shows the top 10 non-core-entering-core roles in order of occurrences in the lexicon.

Roles	Freq
cause	1,454
location	934
destination	140
time	134
source	124
name	80
beneficiary	64
instrument	63
domain	33
extent	32

Table 5: Top 10 Non-core-entering-core Roles

From Table 5, we can see that *cause* is used most frequently, which usually acts as the proto-agent. It shows that concepts which represent the reason of a predicate are very easy to enter the core argument frame of the predicate. *Location* and *time* take second and fourth place. The third and fifth are *destination* and *source*, which often used to represent start and end point of location or time.

4.2 Representation of Multi-functional Roles

Although it is impossible that a predicate has more than 5 core arguments, CPB does not limit the types of concepts that can act as core roles of predicates. As long as it is an indispensable component of the meaning of a predicate, it can act as a core role of the predicate no matter what semantic relationship it has with the predicate. Take 药物缓解疼痛 (the drug relieves the pain) for example, the 药物 (drug) can serve as the agent as well as the cause of the predicate *relieve*, so the concept which represents agent and cause both can serve as the arg0 of 缓解-01 (relieve) in CPB.

缓解-01 (relieve)
arg0: **cause**, **agent**
arg1: theme

Since the description of core roles in CPB lexicon can only explain its relationship with the predicates, we cannot exactly count how many predicate frames have multi-functional core roles. However, data shows that there have been 1,146 senses whose arg0 can be acted by both concepts of agent and cause, accounting for 4.30% of all the senses. It shows that predicates in Chinese having multi-functional roles is common, and the core argument framework of CPB lexicon can represent the multi-functional roles well. That is to say, the CAMR's definition of core roles is reasonable for semantic representation.

5 Discrimination of Non-core Roles of Chinese AMR

In spite of AMR and CAMR has the same core labels as PropBank and CPB, there is a great difference between them for the quantity of non-core role labels. CAMR has 44 non-core role labels (Table 3), which are much more diversified than the 13 non-core role labels in CPB (Table 6). We calculate the using frequency of each non-core role label in CPB corpus and CAMR corpus, showed in Table 6 and Table 7. The mean deviations of them are 7,271.53 and 440.08, respectively. It means that the degree of difference in the using frequency of non-core role labels is much higher in CPB corpus than in CAMR corpus.

Labels	Description	Freq	%
ADV	adverbial	38,262	46.63
TMP	temporal	16,876	20.57
DIS	discourse maker	10,270	12.52
LOC	locative	7,104	8.66
MNR	manner	3,793	4.62
PRP	purpose or reason	2,344	2.86
DIR	direction	874	1.07
CND	condition	864	1.05
TPC	topic	605	0.74
EXT	extent	521	0.63
BNF	beneficiary	470	0.57
FRQ	frequency	49	0.06
DGR	degree	21	0.03

Table 6: Frequencies of Non-core Role Labels in CPB

Label	Freq	%	Label	Freq	%
beneficiary	2,804	19.21	accompanier	41	0.28
mod	2,098	14.38	topic	40	0.27
polarity	1,615	11.07	direction	37	0.25
*aspect	1,432	9.81	*cunit	36	0.25
manner	1,164	7.98	source	32	0.22
mode	1,097	7.52	cost	21	0.14
time	1,045	7.16	destination	18	0.12
degree	1,012	6.93	ord	17	0.12
cause	366	2.51	poss	15	0.10
purpose	362	2.48	unit	14	0.10
location	335	2.30	example	7	0.05
domain	154	1.06	path	6	0.04
duration	146	1.00	medium	2	0.01
instrument	103	0.71	name	1	0.01
frequency	99	0.68	value	1	0.01
compared-to	86	0.59	consist-of	0	0.00
condition	81	0.56	extent	0	0.00
*tense	76	0.52	part-of	0	0.00
range	73	0.50	polite	0	0.00
*perspective	57	0.39	subevent	0	0.00
li	54	0.37	subset	0	0.00
quant	46	0.32	superset	0	0.00

Table 7: Frequency of Non-core Role Labels in CAMR

It is obvious that the 13 non-core role labels of CPB is differ greatly in using frequency and they are too board to distinguish semantic roles of the predicates. From Table 6, we can see that the frequency of using *ADV* is nearly equal to the sum of the frequency of using other 12 labels. This is because they use *ADV* to represent almost all ambiguous semantic relations, such as 不 which means negation, 再 which means repeat, 首次 which represents order. In addition, *TMP* is unable to distinguish concepts of time, duration and time interval. Therefore, the granularity of the non-core role labels in CPB is too coarse, so it is unsuitable for automatic analysis of semantic relations. Nevertheless, setting too many non-core semantic role labels is also hard for semantic analysis, and is a heavy burden for annotators, such as FrameNet. CAMR setting 44 non-core role labels is more suitable and reasonable due to the fact that it has a satisfactory discrimination.

6 AMR's Solution to Dropped Roles

Compared with other methods of meaning representation such as Dependency Graph, a big advantage of AMR is that it allows to re-analyze and add back dropped concepts in the sentences in order to represent the meaning of sentences more completely. For example, the nominal phrase *the injured* drops the agent of the predicate *injure*, AMR can add back a virtual node *person* for the phrase. Take *one of which…* for another example, it drops the noun that appeared in the preceding clause, AMR can add back a *thing* for it. Dropping semantic roles is common in Chinese. According to the statistics, CAMR annotates 1045 tokens of dropped roles for the 5,000 sentences, which cannot be annotated in other SRL resources. 619 of the added concepts have core semantic relation with the predicate, accounting for 59.23% of all the added concepts.

6.1 Adding back Core Roles for Predicates

Core roles of Predicates are of great significance for the meaning of a sentence. We try to explore whether the permission of adding back roles of AMR can help to annotate core roles of predicates more completely by comparing the CAMR corpus with the CPB corpus.

For each sense of each predicate, according to the difference between the quantity of core roles annotated in the corpus and the number of core roles in the predicate framework lexicon, the annotation of core roles can be classified into three categories: all the core roles are annotated (the difference is 0), not all the core roles are annotated (the difference is less than 0), the core roles are more than that in the lexicon (the difference is more than 0). [4] We call them core roles annotated completely, core roles annotated incompletely and the lexicon lack of core arguments, respectively.

We extract predicated frames from the CAMR corpus and the CPB corpus[5] and calculate the difference per sense. Data shows that there are 101326 tokens of senses of predicates in CPB corpus while 19823 tokens in CAMR corpus. Table 8 shows the distribution of quantity of senses in different difference between the quantity of core roles in the two corpus and the lexicon of CPB.

Corpus	Difference (corpus minus lexicon)	-4	-3	-2	-1	0	1	2	Total
CAMR	Tokens of senses	23	272	1,527	6,862	**11,037**	**99**	**3**	19,823
	% of senses	0.12%	1.37%	7.70%	34.62%	**55.68%**	**0.50%**	**0.02%**	100%
CPB	Tokens of senses	344	1,260	10,060	36,539	**52,735**	**383**	**5**	101,326
	% of senses	0.34%	1.24%	9.93%	36.06%	**52.04%**	**0.38%**	**0.00%**	100%

Note: if the predicate in CPB has semantic relations with multiple roles, it just counts as one tokens of sense.

Table 8: Difference between the Quantity of Core Roles in the Two Corpus and the Lexicon of CPB

From Table 8, we can see that the percentage of predicates whose core roles are annotated completely in the CAMR corpus is 3.64 more than the CPB corpus, and the percentage of senses whose core roles are short for the lexicon in CAMR is higher than that in CPB too. But the percentage of senses whose core roles are annotated incompletely is almost lower than the CPB corpus. It means that the AMR can annotate the core roles of predicates more completely. The main reason is that CAMR allows to re-analyze and add back dropped concepts, so that AMR isn't limited in the words of sentences, but can annotate core roles as complete as possible.

The proportion shows that there are also many predicates whose core roles are annotated incompletely. We consider the main reason is that AMR is a method to represent meaning of sentences, not the whole text, so that much information between sentences are missed. In the future, we will attempt to extend the AMR to the text level in order to represent meaning of texts more completely.

6.2 Adding back Dropped Roles of 3 Categories of Special Structures in Chinese

There are quite a few nominal structures dropping core roles of the predicates in Chinese. We choose three categories of special structures in Chinese to analyze: 的 structures, 所 structures and 所…的 structures. The function of adding concepts of AMR can represent their meanings completely. For example, the 的 structure 受伤的 (the injured) drops the agent of the predicate 受伤 (injure), CAMR can add a virtual node *person* and annotate the relationship between the dropped role and the predicate by *person :arg0-of 受伤*. Moreover, it is common that the patient of 的 structures is dropped. Take 我说的 ((what) I said) for example, it drops the theme of 说 (say), CAMR can add a *thing* for the structure. The 所 and the predicate in 所 structures form a nominal structure. Similar to 我说的, 所说 ((what) is said) drops the theme of 说 and CAMR can add back a *thing*. It seems impossible that a 所 structure drops its agent. A 所…的 structure is a combination of a 所 structure and a 的 structure. 所共有的 ((thing) shared by some people) drops a semantic role of 共有 (share), CAMR can also add back a *thing*.

We extracted all the 的, 所 and 所…的 structures in the corpus. According to the statistics, there are 309 的 structures, 9 所 structures and 7 所…的 structures in the 5,000 CAMR sentences. Though not very numerous, they are important and not negligible in Chinese. Data also shows that the number of dropped roles of agent and patient of 的 structures are essentially equal and the most dropped agents are *person* and most dropped patients are

[4] If the difference is less than 0, it is also possible that there are core roles being dropped. Similarly, if the difference is more than 0, it is also possible that there are core roles have not being annotated. But these two cases can be negligible because they are few in number.

[5] Because predicates which do not have core roles in CAMR corpus are difficult to be separated from other words, we ignore them for the moment.

thing. In addition, the dropped roles of these 9 所 structures and 7 所…的 structures are all patients and *thing*. Owing to the scale of the corpus is small, the data may not be able to cover all the situations, but it can also show that the dropped concepts of 所 and 所…的 structures are always the patient of the predicates, which is mainly because these two kinds of structures can represent the objects of actions by themselves.

From the data analysis of the adding semantic roles of predicates and the three types of nominal structures, we can see that the permission of re-analyzing and adding back roles of AMR can help to annotate the meanings of predicates more complete.

The CAMR's function of adding dropped roles also benefits from the framework designed by Li et al. (2017) that can align the AMR concepts and relations to word tokens in a sentence. It uses the index of a word token as the ID of its aligned concept in the AMR representation. When adding a role that is dropped, the added role will be assigned an ID which greater than the length of the sentence. Therefore, it is impossible to confuse the added roles with the words in the given sentence.

7 Conclusion and Future Work

In this paper, based on data analysis of the 5,000 sentences Chinese AMR corpus, we find that the AMR's definition of core roles can solve the collision between core and non-core roles and represent the multi-functional roles well. And the 44 non-core role labels of CAMR have a satisfactory discrimination to non-core semantic roles. In addition, benefited from the permission of re-analyzing and adding concepts, AMR can solve the problem of dropped semantic roles in the sentences, which is especially helpful for annotating special structures in Chinese such as 的 structures. Therefore, as a method of representing meaning of sentences, AMR has unique advantages in semantic role labeling and it is suitable for representing meanings of Chinese sentences, so we need to build a larger AMR corpus to serve the Chinese semantic processing.

A high-quality predicate framework lexicon is significant for ensuring the quality of the annotation. However, there are still many problems in the predicate lexicon we use at present: (1) senses of ambivalent words are not clear; (2) semantic roles do not correspond to the same core arguments, for example the concept of cause is arg0 of 压迫-01 (oppress) and arg1 of 取舍-01 (make the choice); (3) incomplete arguments and senses, for example 贴补-01 (subsidize) is lack of arguments of object and recipient and there is no adjective sense of 丰富 (abundant) in the lexicon. These problems has lowered the quality of annotation and the accuracy of automatic analysis, so we plan to modify the predicate frames manually. Moreover, some nouns have arguments like verbs such as 信心(confidence), but they are not included in the lexicon.

In the future, we will try to annotate semantic roles of nouns in Chinese AMR. And we plan to release our data for NLP applications and linguistics studies.

Acknowledgements

We thank the reviewers. This work is partially supported by projects 61772278, 61472191 under the National Science Foundation of China, project MJUKF201705 under Open Fund Project of Fujian Provincial Key Laboratory of Information Processing and Intelligent Control (Minjiang University), the Project for Jiangsu Higher Institutions' Excellent Innovative Team for Philosophy and Social Sciences, and the Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions.

References

Badarau, B., Bonial, C., Georgescu, M., et al. Abstract Meaning Representation. https://amr.isi.edu/index.html.

Bai, X., & Xue, N. (2016). Generalizing the semantic roles in the Chinese Proposition Bank. *Language Resources and Evaluation*, 50(3), pp. 643-666.

Baker, C. F., Fillmore, C. J., & Lowe, J. B. (1998). The berkeley framenet project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, pp. 86-90.

Banarescu, L., Bonial, C., Cai, S., et al. (2013). Abstract meaning representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pp. 178-186.

Chen, K. J., Luo, C. C., Chang, M. C., et al. (2003). *Sinica Treebank*. Springer Netherlands, pp. 231-248.

Cai, S., & Knight, K. (2012). Smatch: an Evaluation Metric for Semantic Feature Structures. In *Meeting of the Association for Computational Linguistics*, pp.748-752.

Fillmore, C. J., et al. FrameNet. https://framenet.icsi.berkeley.edu/fndrupal/

Jia, J. Z. (2007). Study on the Comparison of Framenet Wordnet Verbnet. *Information Science*. 25(11), pp.1682-1686.

Jurafsky, D., & Martin, J. H. (2014). *Speech and language processing* (Vol. 3). London: Pearson.

Kipper, K., Dang, H. T., & Palmer, M. (2000). Class-Based Construction of a Verb Lexicon. In *Seventeenth National Conference on Artificial Intelligence & Twelfth Conference on Innovative Applications of Artificial Intelligence*, pp. 691-696.

Li, B., Wen, Y., Bu, L., et al. (2017). A Comparative Analysis of the AMR Graphs from English and Chinese Corpus of the Little Prince. *Journal of Chinese Information Processing*, 31(1), pp. 50-57.

Li, B., Wen, Y., Bu, L., et al. (2016). Annotating the Little Prince with Chinese AMRs. In *Proceedings of the 10th Linguistic Annotation Workshop*.

Li, B., Wen, Y., Song, L., et al. (2017). Construction of Chinese Abstract Meaning Representation Corpus with Concept-to-word Alignment. *Journal of Chinese Information Processing*, 31(6), pp. 93-102.

Li, R., Lv, G., Gao, J. et al. Chinese FrameNet. http://sccfn.sxu.edu.cn/portal-zh/home.aspx.

Ma, W., Chen, K., Xie, Y., et al. Sinica Treebank. http://turing.iis.sinica.edu.tw/treesearch/.

Palmer, M., Gildea, D., & Kingsbury, P. (2006). The Proposition Bank: an Annotated Corpus of Semantic Roles. *Computational Linguistics*, 31(1), pp. 71-106.

Palmer, M., Kipper, K. Loper, E, et al. VerbNet. http://verbs.colorado.edu/~mpalmer/projects/verbnet/members.html.

Weischedel, R., Hovy, E., Marcus, M., et al. (2011). Ontonotes: a Large Training Corpus for Enhanced Processing. *Handbook of Natural Language Processing & Machine Translation*.

Xue, N. (2006). A Chinese Semantic Lexicon of Senses and

Roles. *Language Resources & Evaluation*, 40(3-4), pp. 395-403.

Xue, N., Palmer, M. (2009). Adding Semantic Roles to the Chinese Treebank. *Natural Language Engineering*, 15(1), pp. 143-172.

Xue, N., Wang, C., Zhang, Y., et al. Chinese Abstract Meaning Representation. http://www.cs.brandeis.edu/~clp/camr/camr.html.

Xue, N., Xia, F., Chiou, F. D., & Palmer, M. (2005). The Penn Chinese Treebank: Phrase Structure Annotation of a Large Corpus. *Natural Language Engineering*, 11(2), pp. 207-238.

Yuan, Y. L. (2007). The Fineness Hierarchy of Semantic Roles and Its Application in NLP. *Journal of Chinese Information Processing*, 21(4), pp. 10-20.

MALINDO Morph: Morphological dictionary and analyser for Malay/Indonesian

Hiroki Nomoto⋆, Hannah Choi°, David Moeljadi°, Francis Bond°
⋆Tokyo University of Foreign Studies
3-11-1 Asahi-cho, Fuchu, Tokyo 183-8534 Japan
nomoto@tufs.ac.jp
°Nanyang Technological University
14 Nanyang Drive, Singapore 637332
YUNJUNG001@e.ntu.edu.sg, davidmoeljadi@gmail.com, bond@ieee.org

Abstract

Malay/Indonesian lacked an open wide-coverage dictionary that can be used for both NLP tasks and non-NLP purposes. The MALINDO Morph morphological dictionary is the first such dictionary. It provides morphological information (root, prefix, suffix, circumfix, reduplication) for roughly 232K surface forms. The entry forms are those found in the authoritative dictionaries in Malaysia (*Kamus Dewan*[4]) and Indonesia (*Kamus Besar Bahasa Indonesia*[5]) (core dictionary) as well as frequent words in the Leipzig Corpora Collection (Goldhahn et al., 2012) (expanded dictionary). The morphological analyses were checked by hand for all surface forms, except for (i) basic and *di-* forms in the expanded dictionary whose existence is predicted from the corresponding *meN-* active forms in the core dictionary and (ii) the case variants of the items in the core dictionary. This paper also discusses the morphological analyser that we developed to create our morphological dictionary. Our morphological analyser is more linguistically rigorous than previous morphological analysers and stemmers/lemmatizers such as MorphInd (Larasati et al., 2011) because it takes into account circumfixes, which have previously been neglected, largely due to a misunderstanding among NLP researchers that circumfixes are no more than combinations of a prefix and a suffix.

Keywords: Malay/Indonesian, morphological dictionary, morphological analyser

1. Introduction

A good dictionary with wide coverage is crucial to the success of a robust morphological analysis, which in turn becomes the basis for higher-level tasks such as syntactic parsing. While open dictionaries such as the NAIST Japanese Dictionary[1] and UniDic[2] are available for Japanese, nothing comparable exists for Malay/Indonesian. Hence, we created a morphological dictionary for Malay/Indonesian. This paper describes our dictionary and the morphological analyser that we developed for its creation. Both the dictionary and morphological analyser will be made publicly available at `https://github.com/matbahasa/MALINDO_Morph`, licensed under a Creative Commons Attribution 4.0 International (CC BY 4.0) license.

This paper is organized as follows. First, we present a brief overview of the Malay and Indonesian languages (section 2) and their morphology (section 3). Section 4 summarizes previous work on dictionaries for NLP tasks and on stemmers, lemmatizers and morphological analysers for Malay/Indonesian. The tools that we have developed, the MALINDO Morph morphological dictionary and morphological analyser, are described in section 5. Section 6 concludes the paper and discusses ways of using the MALINDO Morph dictionary for NLP and non-NLP purposes. It also suggests ways in which the MALINDO Morph dictionary can be enriched in the future.

2. Malay and Indonesian

The "Malay" language (ISO693-3 msa), from the Austronesian language family, is the official language of four Southeast Asian countries in different parts of the Malay Archipelago. There are two regional varieties of the same language, namely Malay in the narrow sense (ISO693-3 zsm), used in Malaysia, Brunei and Singapore, and Indonesian (ISO693-3 ind), used in Indonesia. In this paper, we refer to the Malay language in the narrow sense simply as "Malay."

Many tools and resources are available that have been independently developed in each region, including standard dictionaries and language resources. In addition, some collaboration has occurred, such as the Majlis Bahasa Brunei-Indonesia-Malaysia (Language Council of Brunei-Indonesia-Malaysia) or MABBIM, a regional language organization whose role is to plan and monitor the development of the Malay/Indonesian language in the region, with Singapore as an observer. While some variations exist between the two languages, they are mutually intelligible, with only about 10% of lexical difference (Asmah, 2001). The two languages also share the same set of affixes. As such, a morphological dictionary can be developed that covers both Malay and Indonesian.

3. Malay/Indonesian Morphology

Malay/Indonesian is an agglutinating language whose morphology involves the use of affixation, reduplication and cliticization.[3] It has productive prefixes, suffixes and cir-

[1] `https://ja.osdn.net/projects/naist-jdic/`

[2] `http://pj.ninjal.ac.jp/corpus_center/unidic/`

[3] A comprehensive description of these processes can be found, among others, in Abdullah (1974), Asmah (2009) and Sneddon et al. (2010).

cumfixes, which can be either derivational or inflectional. It also has infixes, but they are no longer productive. Productive reduplication is achieved through full reduplication of stems (e.g. *kucing* 'cat' → *kucing-kucing* 'cats'). Its semi-productive morphological processes include rhythmic reduplication, which involves vowel and/or consonant alternation (e.g. *gunung* 'mountain' → *gunug-ganang* 'mountain range'). Partial reduplication, which adds the base-initial consonant plus *e* to the base, is semi-productive at best in Indonesian but somewhat productive in Colloquial Malay (e.g. *mula* 'to start' → *memula* 'at first' (= *mula-mula*)). The clitics consist of proclitics (e.g. *ku*= 'I') and enclitics (e.g. =*ku* 'me/my').

The interaction of different morphological processes can give rise to quite a complex word structure. For example, *keterbatasan-keterbatasan* 'limitations' is derived from the root *batas* 'limit', as shown in Figure 1. Notice that the relative order between affixation and reduplication is not fixed. The reverse order is also possible, as illustrated by *keanak-anakan* 'childishness' in the same figure.

4. Existing Tools and Their Problems

4.1. Morphological Dictionary

No large dictionary file is publicly available in an accessible format. The Malay tokenizer/lemmatizer described in Baldwin and Su'ad (2006) has a small dictionary file, which consists of word-lemma-POS (part of speech) triples for 2,499 words.[4]

One can create a larger dictionary by using the data from online dictionaries (not specifically for NLP) such as *Dr. Bahnot's Malay-English Cyber-Dictionary*[5] and *Kateglo ∼ Kamus, tesaurus, dan glosarium bahasa Indonesia*.[6] The latter takes most of its data from the third edition of *Kamus Besar Bahasa Indonesia* and provides an API to access its structured data under a CC BY-NC-SA 3.0 license. However, to the best of our knowledge, no existing dictionary contains the kinds of morphological information that our dictionary offers: affixes (prefixes, suffixes, circumfixes), clitics (proclitics, enclitics) and reduplication types.

4.2. Morphological Analyser

Much work has been done in the past on stemmers/lemmatizers for Malay/Indonesian (see, for example, Baldwin and Su'ad (2006), Adriani et al. (2007), Larasati et al. (2011), Mohamad Nizam et al. (2016) and the studies cited therein). Not mentioned in these papers but notable is the Sastrawi stemmer,[7] which uses Kateglo (see section 4.1) as its dictionary and is offered in multiple languages, namely PHP, Java, C, Python, Go and Ruby.

Morphological analysers analyse the non-stem/lemma strings of a word in addition to identifying the stem/lemma. Currently, MorphInd (Larasati et al., 2011) seems to be the most sophisticated morphological analyser for Malay/Indonesian. It identifies morpheme boundaries and assigns two POS tags to a token: one for the lemma ('lemma tag') and another for the entire token ('morphological tag'). For instance, the verb *mengirim* 'to deliver', which is derived from the root *kirim* by attaching the prefix *meN*-, is analysed as `meN+kirim<v>_VSA`. `<v>` is the lemma tag for verbs, whereas `_VSA` is the morphological tag indicating that the entire token is a singular active verb.[8]

There is a common misunderstanding among NLP researchers about Malay/Indonesian morphology, specifically concerning the notion of the 'circumfix' (also called 'confix'). Circumfixes are incorrectly thought of as a combination of a prefix and a suffix. However, a circumfix is in fact a single morpheme that surrounds a stem. It is true that *meN-X-kan* contains the prefix *meN*- and the suffix -*kan*, but one must not describe *meN- -kan* as a circumfix, as a circumfix encodes syntactic and semantic information that cannot be ascribed to the component parts. The meaning of *ke- -an*, which is a genuine circumfix, cannot be obtained by combining the meanings of *ke*- and -*an*.

Presumably due to this misunderstanding, MorphIndo analyses the non-lemma strings, but it does not specify what they are, that is, whether they are a prefix, suffix or circumfix. For example, *pengiriman* (= *kirim* + circumfix *peN- -an*) 'delivery' is analysed as `^peN+kirim<v>+an_NSD$`. From this output, it is not obvious whether *peN* and *an* are a combination of two morphemes (prefix *peN*- and suffix -*an*) or a single morpheme (circumfix *peN- -an*). In fact, the correct identification of circumfixes presents a major challenge to morphological analysis in Malay/Indonesian.[9] This is because the strings appearing in circumfixes constitute a proper subset of those appearing in prefixes and suffixes. A correct circumfix cannot be identified by just looking at the two strings at the left and right edges of a token. Thus, *berakhiran* 'suffixed' can be segmented as `ber-akhir-an`, but the word does not contain the circumfix *ber- -an*. The word is derived from the root *akhir* by attaching the suffix -*an* to derive *akhiran* 'suffix' and then attaching the prefix *ber*- to this derived form. Likewise, *berperadaban* 'civilized', which is segmented as `ber-per-adab-an`, has a circumfix, but it is not *ber- -an* but *per- -an*.

5. MALINDO Morph

5.1. Morphological Dictionary

Size and format The MALINDO Morph morphological dictionary currently has a total of 232,550 lines, with each containing an analysis for one (case-sensitive) token. These 232,550 tokens are based on 78,750 distinct roots. Each line is made up of the following six items, separated by tabs:

[4] `https://github.com/averykhoo/malay-toklem/blob/master/lexicons/word-lemma-pos`

[5] `http://dictionary.bhanot.net/`

[6] `http://kateglo.com/`

[7] `https://github.com/sastrawi/sastrawi`

[8] Since Malay/Indonesian does not have subject-verb agreement, the number information should in fact be unspecified. Moreover, roots may not need POS tags because roots, unlike surface forms, are abstract entities. This point is clear in languages like Arabic, in which roots are not used as surface forms (e.g. root *k-t-b* 'having to do with writing' → surface forms *kataba* '(he) wrote' (verb), *kitab* 'book' (noun), etc.).

[9] The difficulty involved in distinguishing circumfixes from combinations of a prefix and a suffix has also been noted by Ranaivo-Malançon (2004).

batas	‘limit’	*anak*	‘child’
↓	(affixation: *ter-*)	↓	(reduplication)
terbatas	‘limited’	*anak-anak*	‘children’
↓	(affixation: *ke- -an*)	↓	(affixation: *ke- -an*)
keterbatasan	‘limitation’	*keanak-anakan*	‘childishness’
↓	(reduplication)		
keterbatasan-keterbatasan	‘limitations’		

Figure 1: The derivations of *keterbatasan-keterbatasan* ‘limitations’ and *keanak-anakan* ‘childishness’

- Root
- Surface form
- Prefix(es), proclitic: *meN-*, *N-* (Indonesian), *di-*, *per-*, *ber-*, *ter-*, *peN-*, *pe-*, *ke-*, *se-*; *ku=*, *kau=*[10]
- Suffix(es), enclitic(s): *-kan*, *-i*, *-in* (Indonesian), *-an*, *-nya*; *=ku*, *=mu*, *=kau*, *=nya*, *=lah*, *=kah*
- Circumfix(es): *ber- -an*, *ber- -kan*, *ke- -an*, *peN- -an*, *pe- -an*, *per- -an*, *se- -nya*
- Reduplication: Full, Partial, Rhythmic

Some sample lines are shown in Figure 2.

Our dictionary was built in two steps. First, we built a core dictionary with entries from the authoritative dictionaries in Malaysia and Indonesia, respectively: *Kamus Dewan*[4] (KD) and *Kamus Besar Bahasa Indonesia*[5] (KBBI). Then, we created an expanded dictionary for other tokens that are not listed in KD or KBBI. The source of the expanded dictionary was the reclassified version of the Leipzig Corpora Collection (LCC; Goldhahn et al. (2012); Nomoto et al. (under review)). Tables 1 and 2 summarize the current sizes of the two dictionaries and the frequencies of different morphological processes found in them, respectively.

Dictionary	Checked	Unchecked	Total
Core	84,402	0	84,402
Expanded	47,399	100,749	148,148
Total	131,801	100,749	232,550

Table 1: Sizes of the MALINDO Morph dictionaries (unit: line)

Morphology	Core	Expanded	Total
PREFIXES AND PROCLITICS			
meN-	12,336	8,939	21,275
N-	2	147	149
di-	167	10,787	10,954
per-	634	1,146	1,780
ber-	4,936	3,514	8,450
ter-	2,600	2,190	4,790
peN-	2,127	2,345	4,472
pe-	177	313	490
ke-	123	524	647
se-	1,038	1,621	2,659
ku=	0	1,258	1,258
kau=	0	84	84
SUFFIXES AND ENCLITICS			
-kan	4,423	11,565	15,988
-i	1,390	4,151	5,541
-in	0	25	25
-an	2,718	4,785	7,503
-nya/=nya	70	28,716	28,786
=ku	0	4,209	4,209
=mu	0	2,862	2,862
=kau	0	45	45
=lah	13	6,513	6,526
=kah	5	999	1,004
CIRCUMFIXES			
ber- -an	624	239	863
ber- -kan	227	57	284
ke- -an	2,431	4,146	6,577
peN- -an	2,387	3,040	5,427
pe- -an	76	93	169
per- -an	694	1,357	2,051
se- -nya	92	139	231
REDUPLICATION			
Full	3,693	4,908	8,601
Partial	230	105	335
Rhythmic	735	232	967
NO MORPHOLOGY (ROOT = SURFACE FORM)			
None	50,350	71,264	121,614

Table 2: Morphological profile of the MALINDO Morph dictionaries (unit: token)

Core dictionary We extracted all of the roots and their derived forms from KD and KBBI. We did this manually for KD.[11] As for KBBI, we extracted root and surface forms from a database built by a team that includes the third author of the present paper as a member (Moeljadi et al., 2017).

[10] This slot may also include other items occurring before the root, such as the preposition *ke* ‘to’ as in *mengebumikan* ‘to bury’ or the negator *tidak* ‘not’ as in *ketidakcukupan* ‘insufficiency’.

[11] In fact, the KD database has been commercialized, and we could have purchased the necessary information from a company. However, we did not do so because the price offered was more expensive than the cost of manual work.

Root	Surface form	Prefix	Suffix	Circumfix	Reduplication
perlu	perlu	0	0	0	0
perlu	seperlunya	0	0	se- -nya	0
perlu	memerlukan	meN-	-kan	0	0
perlu	perlu-memerlukan	meN-	-kan	0	R-full
perlu	keperluan	0	0	ke- -an	0

Figure 2: Morphological analysis for the root *perlu* 'necessary' and its derivatives

The morphological analyses were conducted using Microsoft Excel functions. The results were manually checked by Japanese undergraduate students who had learnt Malay or Indonesian as their major for more than three years, Indonesian research students and the first and second authors of the present paper.

For some words, KD and KBBI assume different morphological analyses. There are also cases in which their analyses are good enough for practical purposes but not very precise as linguistic analyses. In such cases, we adopted our own analyses which we think are adequate linguistically. For example, both KD and KBBI list *anai-anai* 'termite' as a head word of its own, whereas the MALINDO Morph morphological dictionary lists the word as a derivative of the root *anai*. KD and KBBI make a practically reasonable choice, given that the form *anai* is a bound root and is not used by itself. However, for a rigorous linguistic analysis, *anai-anai* should be treated as a fully reduplicated form of the bound root *anai*. With departures like this, our core dictionary is not identical to either KD or KBBI.

Expanded dictionary Sixteen 300K subset files (Malay 3, Indonesian 13) of LCC were used as sources of additional data to expand our dictionary. Each 300K file consists of 300,000 sentences. 1,005,007 word types (case-sensitive) were not found in the core dictionary. They include genuine Malay/Indonesian words, proper names, abbreviations, spelling variants/errors, foreign words and non-alphabets. Out of these words, only frequent ones that occurred at least ten times in one of the sixteen subset files were subjected to further processes.

There were 282,186 such words, of which 57,633 were English words and 76,638 were non-alphabets and were not included in the MALINDO Morph morphological dictionary. The remaining 147,915 words were analysed using the morphological analyser described below.[12] The results of the morphological analysis were checked by hand, except for the basic and *di-* forms (cf. Table 3) as well as the case variants of the items in the core dictionary.

The expanded dictionary also contains words in the core dictionary that can also be analysed as involving an enclitic. For example, *masalah* 'problem' and *penanya* 'questioner' are listed in the core dictionary as a root and a *peN-* nominal of the root *tanya* 'ask', respectively. However, they can also be analysed as *masa* 'time' + *-lah* (focus particle) and *pena* 'pen' + *-nya* 'his/her', respectively. These and other analyses were done manually and hence were added to the "checked" category of the expanded dictionary.

Limitations Currently, the MALINDO Morph morphological dictionary only targets productive native affixes and reduplication. This is because they play more important roles compared to non-productive and foreign affixes. Borrowed affixes such as *anti-* 'anti-' and *pra-* 'pre-' are thus not analysed unless they occur together with native ones (e.g. *anti-* in *anti-pemerintah* 'anti-government': perintah anti-pemerintah anti-+peN- 0 0 0).

Moreover, no distinction is made between the suffix *-nya* (forming adverbials, nominalizing verbs and adjectives, occurring in exclamatives) and the enclitic *=nya* (third person pronoun, definite marker). Ideally, the latter should be taken from the word during tokenization. However, tokenizing the enclitic *=nya* without overtokenizing the suffix *-nya* seems almost impossible without referring to a dictionary.

5.2. Morphological Analyser

Preparation First, we built a list of roots (rootlist) and a hypothetical dictionary (hyp-dic) consisting of the basic and *di-* passive forms corresponding to the *meN-* verbs in the core dictionary (core-dic). Basic forms are verb stems without the active voice marker *meN-*. They are used in imperative, bare active and bare passive constructions (see Nomoto (2013) for the voice system in Malay/Indonesian). Table 3 illustrates these three verb forms. Malay/Indonesian dictionaries do not list the basic *di-* forms of a verb.[13] However, they can be created automatically by removing the prefix *meN-* from the *meN-* form (basic form) and prefixing *di-* to the resulting form (*di-* form), respectively. The forms thus created are merely hypothetical. Hence, they were added to the expanded dictionary (exp-dic) only if they were found to actually be used in the corpus.

The algorithm Given input W, our morphological analyser works as follows. The 'analysis' in Steps 1–5 is a list of the format ⟨ affix candidate, root, remaining string before root, remaining string after root, reduplication ⟩.

Step 1. If W is a non-alphabet, return analysis $\langle \emptyset, W, \emptyset, \emptyset, \emptyset \rangle$.

Step 2. If W or its lowercase equivalent w is an English word, return analysis $\langle \emptyset, W/w, \emptyset, \emptyset, \emptyset \rangle$.

Step 3. If W/w is in core-dic/hyp-dic, retrieve the line(s) for W/w in core-dic/hyp-dic.

[12] This number is smaller than that reported in Table 1 above because some words are morphologically ambiguous, with two or more possible analyses.

[13] The exceptions include Asakura (1963), Quinn (2001), Nomoto (2016) and Florentina (2017).

Basic (= stem)	*MeN-* active	*Di-* passive	Common morphology
ajar 'to teach'	*mengajar* 'to teach'	*diajar* 'to be taught'	Root *ajar*
ajarkan 'to teach (for)'	*mengajarkan* 'to teach (for)'	*diajarkan* 'to be taught (for)'	Root *ajar* + suffix *-kan*
pelajari 'to learn'	*mempelajari* 'to learn'	*dipelajari* 'to be learnt'	Root *ajar* + prefix *per-* + suffix *-i*

Table 3: Verbal inflection in Malay/Indonesian

Step 4. Strip W/w of clitic strings. If the resulting form r is in core-dic/hyp-dic, retrieve the line(s) for r in core-dic/hyp-dic and add the clitic information.

Step 5. Generate candidate sets $Cand_c$, $Cand_p$ and $Cand_s$, where $Cand_a$ is a set of candidate analyses for token w based on affix/clitic type $a \in \{c(\text{ircumfix}), p(\text{refix/proclitic}), s(\text{uffix/enclitic})\}$.

Step 6. Search the direct product $Cand_c \times Cand_p \times Cand_s$ for members whose elements are mutually compatible.

Step 7. Return $\langle root_c, w, p\text{-}, \text{-}s, c_1\text{- -}c_2, red_c \rangle$ for every such member.

The notion of mutual compatibility among analyses invoked in 6 is defined as follows:

Definition Three lists, $\langle c_1\text{- -}c_2, root_c, start_c, end_c, red_c \rangle$, $\langle p\text{-}, root_p, start_p, end_p, red_p \rangle$ and $\langle \text{-}s, root_s, start_s, end_s, red_s \rangle$, are mutually compatible if and only if all of the conditions below are satisfied:

1. $root_c = root_p = root_s$
2. $red_c = red_p = red_s$
3. $start_c = p$
4. $start_p = c_1$
5. $start_s = c_1 + p$
6. $end_c = s$
7. $end_s = c_2$
8. $end_p = c_2 + s$

Example Let us consider an example from core-dic: *sedianya* 'actually'. The word is made up of the root *sedia* and the suffix *-nya*. Suppose that this form did not exist in core-dic. The word is neither a non-alphabet (Step 1) nor an English word (Step 2. It is not in hyp-dic (Step 3). It contains a clitic string, namely *nya*. So, by Step 4, we remove *nya* and check the dictionaries to determine whether the remaining string *sedia* exists in them. It actually does; core-dic has this line: `sedia sedia 0 0 0 0`. Incorporating the clitic information into this, our morphological analyser will return $\langle \text{sedia}, \text{sedianya}, \emptyset, \text{-nya}, \emptyset, \emptyset \rangle$. It is a correct result, although the distinction between the suffix *-nya* and the clitic *=nya* has been lost.[14]

To see how Steps 5–7 work, let us suppose that Step 4 failed for some reason. The candidate sets obtained by Step 5 are:

$$Cand_c = \left\{ \begin{array}{l} \langle \emptyset, \text{sedia}, \emptyset, \text{nya}, \emptyset \rangle, \langle \emptyset, \text{dia}, \text{se}, \text{nya}, \emptyset \rangle, \\ \langle \text{se- -nya}, \text{dia}, \emptyset, \emptyset, \emptyset \rangle \end{array} \right\}$$

$$Cand_p = \left\{ \begin{array}{l} \langle \emptyset, \text{sedia}, \emptyset, \text{nya}, \emptyset \rangle, \langle \emptyset, \text{dia}, \text{se}, \text{nya}, \emptyset \rangle, \\ \langle \text{se-}, \text{dia}, \emptyset, \text{nya}, \emptyset \rangle \end{array} \right\}$$

$$Cand_s = \left\{ \begin{array}{l} \langle \emptyset, \text{sedia}, \emptyset, \text{nya}, \emptyset \rangle, \langle \emptyset, \text{dia}, \text{se}, \text{nya}, \emptyset \rangle, \\ \langle \text{-nya}, \text{sedia}, \emptyset, \emptyset, \emptyset \rangle, \langle \text{-nya}, \text{dia}, \text{se}, \emptyset, \emptyset \rangle \end{array} \right\}$$

Step 6 picks out the following two lists of lists, based on which Step 7 yields the outputs shown to the right of "→":

1. $$\left(\begin{array}{l} \langle \emptyset, \text{sedia}, \emptyset, \text{nya}, \emptyset \rangle, \langle \emptyset, \text{sedia}, \emptyset, \text{nya}, \emptyset \rangle, \\ \langle \text{-nya}, \text{sedia}, \emptyset, \emptyset, \emptyset \rangle \end{array} \right)$$
   $\to \langle \text{sedia}, \text{sedianya}, \emptyset, \text{-nya}, \emptyset \rangle$

2. $$\left(\begin{array}{l} \langle \text{se- -nya}, \text{dia}, \emptyset, \emptyset, \emptyset \rangle, \langle \emptyset, \text{dia}, \text{se}, \text{nya}, \emptyset \rangle, \\ \langle \emptyset, \text{dia}, \text{se}, \text{nya}, \emptyset \rangle \end{array} \right)$$
   $\to \langle \text{dia}, \text{sedianya}, \emptyset, \emptyset, \text{se- -nya}, \emptyset \rangle$

Root identification The generation of candidate sets in Step 5 requires root identification. Our morphological analyser contains a root identification function.[15]

Figure 3 shows our root identification algorithm. It is in fact not a simple root identifier; it also identifies reduplication types and pre- and post-root strings. These pieces of information as well as the root information are used for candidate generation in Step 5.

The Hyphen Handler in the algorithm deals with reduplication and other forms with a hyphen, which include hyphenated words (e.g. *Indo-nesia*) and derived words with numeral bases (e.g. *ke-19* '19th', *1990-an* '1990s'). The algorithm of the Hyphen Handler is given in Figure 4.

The "recover root-initial consonant" process recovers a root-initial consonant that does not occur in the surface form as a result of the morphophonological process called 'nasal substitution'. Nasal substitution changes the phone *N* in an affix to a sound that is homorganic to the following consonant, triggering coalescence of the two sounds for native roots starting with a voiceless consonant, as shown in Figure 5.[16] Two outputs are returned: roots with and without a recovered consonant. The Root Well-Formedness Filter filters out items that do not conform to the template for legitimate roots in the language, such as roots with no vowel and roots starting with a complex onset such as *rt*.

[14] We assume that the relevant distinction is handled by a tokenizer.

[15] Alternatively, one can also use existing stemmers/lemmetizers (cf. section 4) for root identification. This is because they normally equate stems/lemmas with roots, even though, strictly speaking, they are distinct units (cf. Table 3).

[16] Coalescence sometimes also occurs with a voiced consonant. See Nomoto (2012) for variations in nasal substitution.

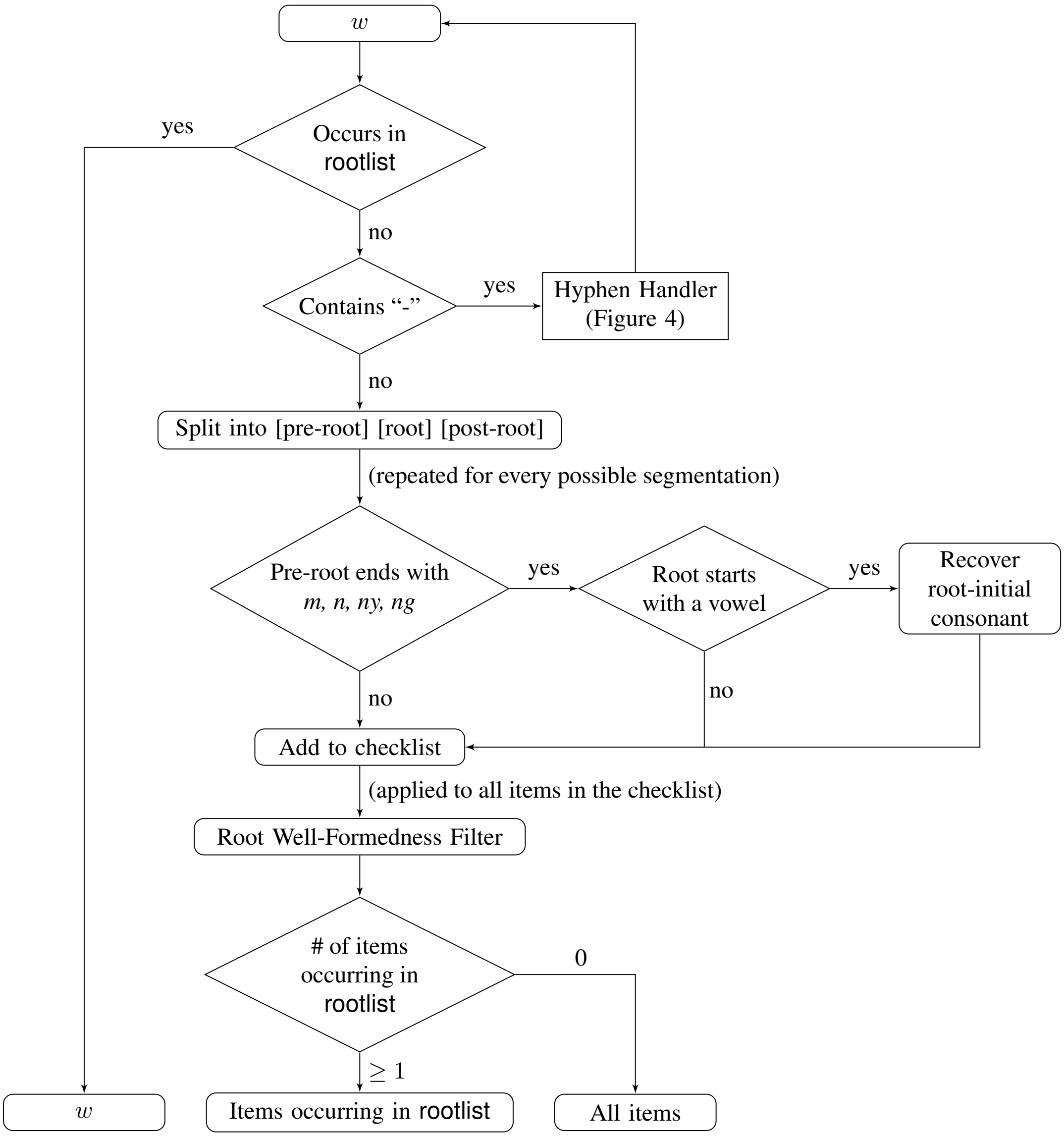

w
Occurs in rootlist
yes
no
Contains "-"
yes
Hyphen Handler (Figure 4)
no
Split into [pre-root] [root] [post-root]
(repeated for every possible segmentation)
Pre-root ends with m, n, ny, ng
yes
Root starts with a vowel
yes
Recover root-initial consonant
no
no
Add to checklist
(applied to all items in the checklist)
Root Well-Formedness Filter
# of items occurring in rootlist
0
≥ 1
w
Items occurring in rootlist
All items

Figure 3: Root identification algorithm

In Figure 6, the root identification algorithm described above is exemplified by the word *mengada-ngadakan* 'to concoct' (= root *ada* + prefix *meN-* + suffix *-kan* + full reduplication). It is a variant of *mengada-adakan* that is not found in either KD or KBBI and hence does not occur in core-dic.

Notice that our root identifier and morphological analyser may return multiple outputs. This is a welcome result because a form can indeed be ambiguous in terms of its morphological composition. For example, a good morphological analyser should be able to come up with the following three analyses for *beruang* in Indonesian:[17]

1. Root *beruang* + no affix	'bear (animal)'
2. Root *ruang* + prefix *ber-*	'to have room'
3. Root *uang* + prefix *ber-*	'to have money'

The disambiguation of multiple morphological analyses is only possible when a concrete context is given. Hence, this is a task for a higher level.

6. Conclusions and Future Work

The MALINDO Morph morphological dictionary, with a total of 232,550 lines, will improve the accuracy of stemming/lemmatization in Malay/Indonesian. Stemming/lemmatizing frequent words will become a simple dictionary lookup with an additional disambiguation process for words with ambiguous analyses. The development of stemmers, lemmatizers and root identifiers should then focus on infrequent words. A possible next step to improve them is to incorporate a spell checker, a named entity recognizer and foreign word identifier. As we manually checked the results of the morphological analysis with our morphological analyser (cf. section 5.1), it turned out that infrequent words are often spelling variants/mistakes,

[17] The third analysis is irrelevant in Malay.

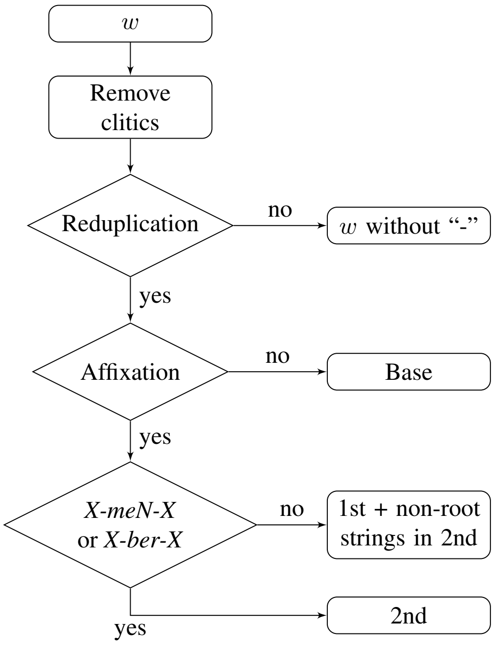


Figure 4: Hyphen Handler algorithm

N-C		Homorganic N		Coalescence
Np	→	mp	→	m
Nb	→	mb	→	m
Nf	→	mf	→	m
Nv	→	mv	→	m
Nt	→	nt	→	n
Nd	→	nd	→	n
Ns(y)	→	ns(y)	→	ny
Nz	→	nz	→	ny
Nc	→	nc	→	ny
Nj	→	nj	→	ny
Nk(h)	→	ngk(h)	→	ng
Ng(h)	→	ngg(h)	→	ng
Nh	→	ngh	→	ng

Figure 5: Nasal substitution

proper names and foreign words, but not a complex combination of productive morphological processes based on known roots.

Furthermore, the MALINDO Morph morphological dictionary provides useful information for other tasks. Parts of speech can be partly predicted from the outermost affix of a word: *meN-* → verb (active), *per- -an* → noun, *se- -nya* → adverb, etc. Specific affixes also provide information about semantics and the argument structure. Words with *peN-* and *peN- -an* are all nouns. However, while the former refers to the external argument (e.g. agent) of the corresponding verb and hence can only take an internal argument (e.g. patient), the latter denotes an action and can take both the external and internal arguments (Nomoto, 2017).

The MALINDO Morph morphological dictionary can also be used for linguistic research. Discoveries of new generalizations regarding the morphological patterns in the language can make a morphological analyser more efficient. The order in which morphological rules are applied is not random. Abdullah (1974) tried to reveal the interaction patterns among them based on 5,153 distinct roots. For instance, he put forward the generalization that "no constructions exceed three layers of affixation" (p. 44). However, no attempt, to our knowledge, has been made to either verify or modify his model based on KD/KBBI-size data using modern computational power.

In the future, the MALINDO Morph dictionary can be enriched by adding more linguistic information. Firstly, the distinction between the suffix *-nya* and the enclitic *=nya* needs to be made in some way (cf. "Limitations" in section 5.1). As a clitic, the latter, by definition, attaches to virtually anything. By contrast, the distribution of the suffix *-nya* is more restricted, though still very wide: (i) it occurs in adverbials (e.g. *biasanya* 'usually' cf. English *-ly*); (ii) it nominalizes verbs and adjectives (e.g. *adanya X* 'the presence of X', cf. Japanese *koto*); (iii) it occurs in exclamatives in Malay (e.g. *sedapnya* 'how delicious(!)', cf. Japanese *koto*). The best way of making this distinction is to flag potential instances of non-suffix *nya*.

Secondly, the information about the variety, i.e. Malay, Indonesian and their dialects, can be added. This can be done by checking whether each word occurs in a corpus of a particular variety. In addition, both KD (Malaysia) and KBBI (Indonesia) contain words that are primarily used in the other variety and indicate such words with special abbreviations. This information can definitely be utilized. However, it must be verified by corpus data before being added to the MALINDO Morph morphological dictionary. The variety information will help to determine a more accurate rate of lexical difference between Malay and Indonesian.

Finally, POSs are another element that a dictionary should provide. As stated above, POSs in Malay/Indonesian can be predicted by morphology, although only partially. Given the large number of lines involved, the POS annotation of the MALINDO Morph morphological dictionary will have to rely on a POS-tagged corpus (ideally the same LCC data that we used) generated by a good POS tagger, the development of which, we believe, should benefit considerably from the MALINDO Morph morphological dictionary.

7. Acknowledgements

The research reported in this paper was conducted under the JSPS grant "Program for Advancing Strategic International Networks to Accelerate the Circulation of Talented Researchers" offered to Tokyo University of Foreign Studies for a project entitled "A Collaborative Network for Usage-Based Research on Less-Studied Languages" as well as the JSPS Grant-in-Aid for Young Scientists (B) (#26770135). We are grateful to JSPS and Nanyang Technological University (NTU) for supporting the first author's stay at NTU.

8. Bibliographical References

KBBI^3 (2001). *Kamus Besar Bahasa Indonesia*. Balai Pustaka, Jakarta, 3rd edn.

KBBI^5 (2016). *Kamus Besar Bahasa Indonesia*. Badan Pengembangan dan Pembinaan Bahasa, Jakarta, 5th edn.

KD^4 (2005). *Kamus Dewan*. Dewan Bahasa dan Pustaka, Kuala Lumpur, 4th edn.

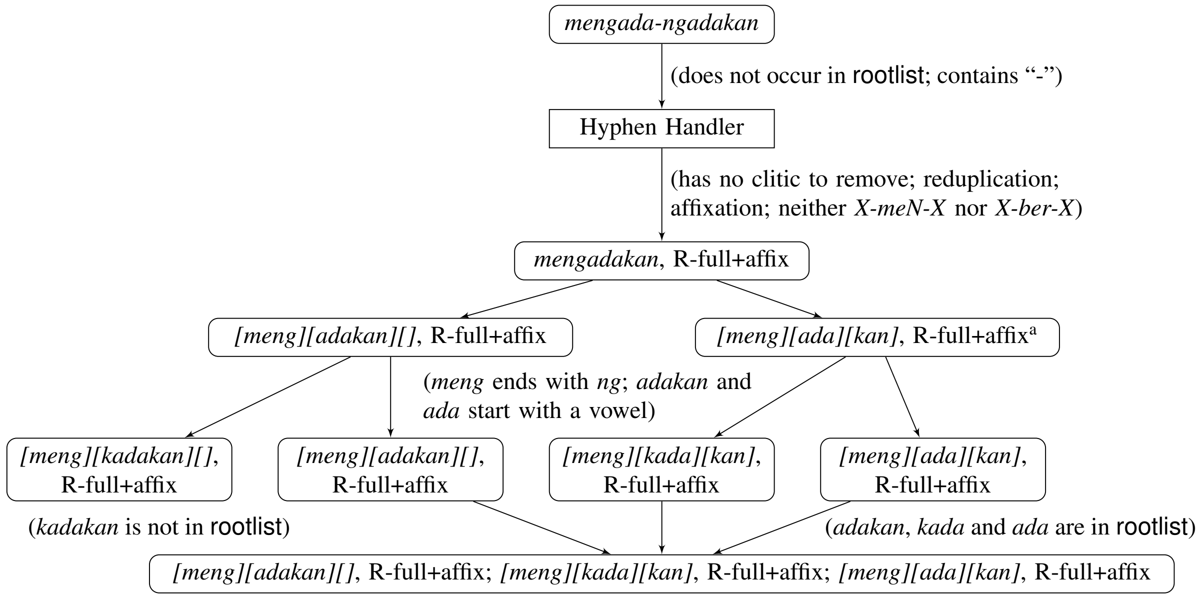

mengada-ngadakan
(does not occur in rootlist; contains "-")
Hyphen Handler
(has no clitic to remove; reduplication; affixation; neither *X-meN-X* nor *X-ber-X*)
mengadakan, R-full+affix
[meng][adakan][], R-full+affix
[meng][ada][kan], R-full+affix[a]
(*meng* ends with *ng*; *adakan* and *ada* start with a vowel)
[meng][kadakan][], R-full+affix
[meng][adakan][], R-full+affix
[meng][kada][kan], R-full+affix
[meng][ada][kan], R-full+affix
(*kadakan* is not in rootlist)
(*adakan*, *kada* and *ada* are in rootlist)
[meng][adakan][], R-full+affix; *[meng][kada][kan]*, R-full+affix; *[meng][ada][kan]*, R-full+affix

Figure 6: Root identification for *mengada-ngadakan* 'to concoct'

[a] *[][mengadakan][]*, *[][mengada][kan]*, *[me][ngadakan][]* and *[me][ngada][kan]* are also possible but omitted here due to space constraints, as they do not affect the final output.

Abdullah, H. (1974). *The Morphology of Malay*. Dewan Bahasa dan Pustaka, Kuala Lumpur.

Adriani, M., Asian, J., Nazief, B., Tahaghoghi, S. M., and Williams, H. E. (2007). Stemming Indonesian: A confix-stripping approach. *ACM Transactions on Asian Language Information Processing (TALIP)*, 6(4):1–33.

Asakura, S. (1963). *Daigakushorin Indoneshiago Shoujiten (Daigakushorin Kamus Bahasa Indonésia-Djepang, Djepang-Indonésia)*. Daigakushorin, Tokyo.

Asmah, H. O. (2001). The Malay language in Malaysia and Indonesia: From lingua franca to national language. *The Aseanists ASIA*, II.

Asmah, H. O. (2009). *Nahu Melayu Mutakhir*. Dewan Bahasa dan Pustaka, Kuala Lumpur, 5th edn.

Baldwin, T. and Su'ad, A. (2006). Open source corpus analysis tools for Malay. In *Proceedings, the 5th International Conference on Language Resources and Evaluation (LREC2006)*, pages 2212–2215.

Florentina, E. (2017). *Pootaburu Nichi-Indoneshia-Ei, Indoneshia-Nichi-Ei Jiten*. Sanshusha, Tokyo.

Goldhahn, D., Eckart, T., and Quasthoff, U. (2012). Building large monolingual dictionaries at the Leipzig Corpora Collection: From 100 to 200 languages. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*.

Larasati, S. D., Kuboň, V., and Zeman, D. (2011). Indonesian morphology tool (MorphInd): Towards an Indonesian corpus. In Cerstin Mahlow et al., editors, *Systems and Frameworks for Computational Morphology*, pages 119–129. Springer, Verlag.

Moeljadi, D., Kamajaya, I., and Amalia, D. (2017). Building the Kamus Besar Bahasa Indonesia (KBBI) database and its applications. In Hai Xu, editor, *Proceedings of the 11th International Conference of the Asian Association for Lexicography*, pages 64–80.

Mohamad Nizam, K., Mohd Aizaini, M., Anazida, Z., and Amirudin, A. W. (2016). Word stemming challenges in Malay texts: A literature review. In *2016 4th International Conference on Information and Communication Technology (ICoICT)*, pages 1–6, May.

Nomoto, H., Akasegawa, S., and Shiohara, A. (under review). Reclassification of the Leipzig Corpora Collection for similar languages: Malay and Indonesian.

Nomoto, H. (2012). More on Austronesian nasal substitution. In M. Ryan Bochnak, et al., editors, *Proceedings from the 45th Annual Meeting of the Chicago Linguistic Society*, volume 1, pages 503–517, Chicago, IL.

Nomoto, H. (2013). On the optionality of grammatical markers: A case study of voice marking in Malay/Indonesian. In Alexander Adelaar, editor, *Voice Variation in Austronesian Languages of Indonesia*, volume 54 of *NUSA*, pages 121–143.

Nomoto, H. (2016). *Pootaburu Nichi-Maree-Ei, Maree-Nichi-Ei Jiten*. Sanshusha, Tokyo.

Nomoto, H. (2017). Sintaksis nominalisasi bahasa Melayu. [The syntax of Malay nominalization]. In Rogayah Abd. Razak and Radiah Yusoff, editors, *Aspek Teori Sintaksis Bahasa Melayu*, pages 71–117. Dewan Bahasa dan Pustaka, Kuala Lumpur.

Quinn, G. (2001). *The Learner's Dictionary of Today's Indonesian*. Allen & Unwin, Sydney.

Ranaivo-Malançon, B. (2004). Computational analysis of affixed words in Malay language. Paper presented at the 8th International Symposium on Malay/Indonesian Linguistics (ISMIL).

Sneddon, J. N., Adelaar, A. K., Djenar, D. N., and Ewing, M. (2010). *Indonesian: A Comprehensive Grammar*. Routledge, London, 2nd edn.

Morphological and Orthographic Challenges in Urdu Language Processing: A Review

Tayyaba Fatima, Raees Ul Islam, Muhammad Waqas Anwar
Department of Computer Science, COMSATS Institute of Information Technology, Lahore, Pakistan
{tayybafatimah, rtraees}@gmail.com, waqasanwar@ciitlahore.edu.pk

Abstract

Urdu is the national language and lingua franca of Pakistan. It is a grammar enriched language. It has a big variety of derivation and inflections in a single word that makes it a challenging language to work on language processing tasks. Research in Natural Language Processing (NLP) and Computational Linguistics (CL) has composed a considerable measure about the history of Urdu language, evolution of Urdu literature, usage of Urdu language in a wide range, effects of other languages on Urdu, Urdu dialect and script etc. Most of the work done on Urdu in the field of Natural Language Processing (NLP) and Computational Linguistics (CL) is related to its morphology, orthography and script. Urdu has a very rich and complex morphology which makes it a challenging language in Natural Language Processing (NLP) and Computational Linguistics (CL) tasks. The purpose of this paper is to comprehensively review the morphological and orthographic challenges that arise in Urdu language processing. In modern linguistics morphology and orthography has a central place. Other branches like historical linguistic, phonemics and morphonemics are also important. But this work focuses on Urdu morphology and orthography. Few studies highlighting these Morphological and Orthographic challenges in Urdu Language Processing (ULP) can be found in the literature but still there are many unsolved problems that need to be highlighted and solved. This paper is intended to present, group, and review these challenges.

Keywords: Natural language processing, Urdu language, Morphology, Orthography, Resources

1 Introduction

Natural Language Processing has become pretty mature in western languages. But in the case of South Asian languages like Urdu, core linguistic resources e.g. corpora, WordNet, dictionaries, tag sets and associated tools are still not available. Urdu has a complex morphology and orthography. There are very few working Natural Language Processing (NLP) applications of Urdu language available. The main reason is that it is a very difficult language to perform any computational task on it. Another big reason is the limited market and relaxed enforcement of copyright laws in the region, which causes unwillingness of the developers towards developing resources and products for language. While working on Urdu language processing, a lot of challenges arises. Whether the work is being done on speech or it is on text, it becomes a challenge to work on Urdu language processing. Some of these challenges are described in the literature, while many others need to shed light on them. In this paper, we have presented the challenges that are faced while working on Urdu language processing. A detailed analysis has been made to dig out these problems and challenges.

2 Previous Work

Some challenges in Urdu language processing have been discussed in the literature. A comprehensive review of different algorithms and techniques of stemming Urdu text is provided by (Abdul Jabbar and Khan, 2016). They discussed the challenges influencing the stemming process because of morphological richness of Urdu language and inclusion of loan words of languages like Hindi, Arabic, Persian etc. They also display an extensive audit of various algorithms and systems of stemming Urdu text and furthermore, investigated about the language structure, morphological closeness and other basic highlighting and stemming approaches utilized as a part of Urdu-like languages i.e. Arabic and Persian. Moreover, they investigated benefits and weaknesses of the utilized stemming approaches in Urdu text stemming. (Ali Daud, 2016) had provided a comprehensive survey on Urdu language processing. The center target of the study is to provide an overview with respect to various linguistics assets that exits for Urdu language processing. The study endeavors to portray in detail the current advances made in Urdu language processing. At first, the accessible data-sets for Urdu language are discussed. Resource sharing amongst Hindi and Urdu, morphology and orthography of Urdu language are discussed. Parts of the text preprocessing tasks such as stop words expulsion, diacritics evacuation, normalization and stemming are delineated. A survey of language processing tasks for example tokenization, Sentence Boundary Disambiguation (SBD), Part of Speech (POS) labeling, Named Entity Recognition (NER), parsing and advancement of WordNet undertakings are talked about. Moreover, effect of Urdu language processing on application regions for example Information Retrieval (IR), classification and plagiarism detection is researched. At last, open issues furthermore, future bearings for this new and dynamic territory of research are given. The objective of their work is to sort out the Urdu language processing work in a way that it can give a stage to research work in Urdu language processing. (Riaz, 2007) has pointed out challenges in writing Urdu stemmer. (Hussain, 2007) has discussed Computational Linguistics (CL) issues in Pakistan and gave some proposals. The study briefly describes the present work on computational linguistics in Pakistan, issues in its improvement and a few propositions for quickening the current pace of work in computational modeling of Pakistani Languages. (Sajjad Ahmad, 2011) presented issues pertaining to the development of a rule based stemmer for Urdu language. They talked about some rule based English, Arabic, Persian and Urdu stemmers. Extremely less work has been done on Urdu stemmer due to its complex and rich morphology. Other than its own vocabulary, Urdu is likewise affected by other morphologies for example Arabic, Persian, Hindi, English and so on. The paper also called attention to a few challenges relating to the advancement of a Urdu stemmer. These issues ought to be considered while building up a rule based Urdu stemmer. (Zobia Rehman, 2001) discussed Challenges in Urdu text tokenization and sentence boundary disambiguation. They divided tokenization issues into two categories: Space inclusion issues and Space exclusion issues. (Waqas Anwar, 2006) carried out a survey about automatic Urdu Language Processing (ULP). It contains the initial attempts in the development of resources for Urdu language processing. They also presented different type of Linguistics analysis in their work.

3 Challenges in Urdu Language Processing

Urdu language has always proved to be a big challenge in the field of language processing. In this section, we discuss the morphological and orthographic challenges of Urdu language processing that are still unsolved.

3.1 Morphological Richness

Natural language processing and computational linguistics tasks become harder as the morphology of the language becomes more complex. Same is the case with Urdu language. It is a morphologically complex language. There exist a number of variants in Urdu against a single word (Abdul Jabbar and Khan, 2016). It is a rich language in the case of both inflectional and derivational morphologies. In order to work on Urdu language processing and its related areas, one should be very clear about its morphology and morphological system.

3.2 Lack of Standardization

A significant problem faced by researchers of Urdu language is lack of standardization of language writing rules. Still there are no standards about writing Urdu text, for example space inclusion and exclusion issues (Zobia Rehman, 2001) i.e. where to add space between words, how to write joiners and non joiners, no standards about writing compound words, what should be added between non-joiners and compound words i.e. space, zero width non-joiner, no-break-space, zero-width space or zero-width no-break space. Lack of standardization regarding Urdu text writing creates serious issue while doing language processing tasks on Urdu language. These issues of writing Urdu text can be solved if proper standards are made and imposed. Challenges created by such issues are highlighted in the following section.

3.3 Text Tokenization and Language Modeling

Lack of standardization in writing text creates a lot of difficulties in text tokenization and Language Modeling (LM). It must be decided that how the compound words will be treated in the Language Modeling (LM) i.e. Unigram, bigram or trigram. Urdu compound words consist of two or three meaningful words. It must be decided that how these words will be treated; whole word as a unigram or first word, (و) (*wao*) and second word as three separate words. It can cause problem while performing text preprocessing tasks like tokenization and stemming. In the following section we have discussed the categories of Urdu compound words in detail.

3.3.1 Compound Words

In Urdu language, when two or more stems combine to make a single meaningful stem, it is called a compound word (مرکب لفظ) (*murakkab-lafZ*). These words are often used to create preciosity and stress in words. Following are the major categories of Urdu Compound words (Zobia Rehman, 2001) (Schmidt, 1999):

1. AB Compound
2. A-o-B Compound
3. A-e-B Compound
4. A-al-B Compound

AB Compound This is a compound in which two stems are attached to make a single stem (Zobia Rehman, 2001). Both of the stems are meaningful but second stem is extra in need. This is called (تابع موضع) (*tabE-moZ-u*). Examples of AB formation are given in Table 1. There is also a situation where first stem is meaningful but second stem does not bear any meaning but still attached to first stem to make a compound. This is called (تابع مہمل) (*tabE-mohmil*). Few examples of such words are given in Table 2.

آس پاس	(*Aas-pas*) (around)
آگے پیچھے	(*Aagy-peecHy*) (back and forth)
صاف سُتھرا	(*Saf-sutHra*) (neat and clean)
چوری چھپے	(*cHoree-cHupay*) (sneak)
رونا دھونا	(*rona-dHona*) (melodrama)
کھانا پینا	(*kHana-pina*) (eat and drink)
آنا جانا	(*Ana-jana*) (come and go)
لینا دینا	(*laina-daina*) (give and take)
چلنا پھرنا	(*cHalna-phirna*) (to walk)
اِدھر اُدھر	(*idhar-udhar*) (here and there)

Table 1: Examples of AB Compound 1

روٹی وٹی	(*roTee-woTee*) (bread)
میل جول	(*mail-jol*) (interaction)
میلا کچیلا	(*maila-kucHaila*) (unclean)
کوڑا کرکٹ	(*kuRa-kirkiT*) (garbage)
خالی خولی	(*kHalee-kHulee*) (empty)
جھوٹ موٹ	(*jhoT-moT*) (lie)
پانی وانی	(*panee-wanee*) (water)
گول مٹول	(*mela-kucHela*) (gol-maTol)
سودا سلف	(*soda-salf*) (Grocery)
کالا کلوٹا	(*kala-kaloTa*) (Black)

Table 2: Examples of AB Compound 2

A-o-B Compound The compound word can be in A-o-B formation (Schmidt, 1999) (Zobia Rehman, 2001). Two words are linked together by the morpheme (و) (*wao*) making a single meaningful word. Few examples of A-o-B formation are given in the Table 3. In these examples, the word Wao (و) is called (حرفِ عطف) (*harf-e-ETf*). It links two stems making the compound word. It gives the meaning of (اور) (*aur*) (and) between two stems.

شب و روز	(*sHab-o-roz*) (day and night)
نظم و نسق	(*naZm-o-nasQ*) (discipline)
ملک و قوم	(*mulk-o-Qom*) (country and nation)
لیل و نہار	(*lail-o-nahar*) (night and day)
خیر و شر	(*khair-o-shar*) (good and bad)
وسیع و عریض	(*wasee-o-Areez*) (wide)
غرور و تکبر	(*gharoor-o-takkabur*) (pride)
قرب و جوار	(*qurb-o-jawar*) (near and around)
صبح و شام	(*subh-o-shaam*) (morning and evening)
مال و متاع	(*maal-o-mataa*), (assets)

Table 3: Examples of A-o-B Compound

A-e-B Compound In this formation, two stems are attached together by writing ◌ِ (*Zair*) under the last letter of first stem. This makes them a single semantic unit. Some examples of A-e-B formation are given in Table 4.

خدمتِ خلق	(*kHidmat-E-kHalq*) (social welfare)
صبحِ صادق	(*subh-E-kaazib*) (early morning)
اہلِ بیت	(*ehl-E-bait*) (people of the house)
دلِ مضطر	(*dil-E-muZta*) (worried heart)
خانہِ خدا	(*kHana-E-kHuda*) (God's house)
وقتِ رخصت	(*waQt-E-rukHsat*) (leaving time)
براہِ مہربانی	(*barah-E-meharbani*) (kindly)
راہِ راست	(*raah-E-raast*) (straight path)
افواجِ پاکستان	(*afwaj-e-Pakistan*) (Pakistan forces)
صدرِ مملکت	(*sadr-E-mumlikat*) (The President)

Table 4: Examples of A-e-B Compound

A-al-B Compound Sometimes two stems are joined together to make a single stem by a joining morpheme (ال) (*al*) . These types of compound words are one of the basic structures of the Arabic root. Table 5 shows few examples of A-al-B compounds.

بین الاقوامی	(*bain-al-aQwamee*) (international)
باب المدینہ	(*baab-al-madeena*) (door of Madina)
ردالفساد	(*radd-al-fasaf*) (elimination of discord)
بیت المقدس	(*bait-al-muQaddas*) (holy house)
ضرب المثال	(*Zarb-al-amsaal*) (idioms)
دین الحق	(*deen-al-HaQ*) (true religion)

Table 5: Examples of A-al-B Compound

3.3.2 Joiners and Non-joiners

There are two types of characters in Urdu: 1) Joiners 2) Non-Joiners.

Joiners In Urdu language, some alphabets are connected with their preceding alphabet. These are called Joiners. List of all Urdu Joiners is given here. Each joiner has three different shapes depending upon its position in word. These three positions may be beginning of word (initial), middle (medial) of word and end of word (final) (Abdul Jabbar and Khan, 2016) (Zobia Rehman, 2001). Three shapes of a joiner (م) (*meem*) are illustrated in Table 7. If a joiner occurs at the end of a word, it is necessary to separate it from next word by adding space. If space is omitted, it will join both words with each other and gives wrong meaning. Here we take the example of joiner (گ) (*gaaf*). In the sentence (آگ جلادو) (*aag jala do*) (turn the fire on), each word is separated with a space. But if we write them without space, they join each other making the sentence wrong like shown in Table 8. The word (fire) (آگ) (*aag*) ends with a joiner (گ) (*gaaf*) which connects itself to the first letter of the next word, if space is not used between them this makes it a wrong word. So space is required between joiners.

ب پ ت ٹ ث ج چ ح خ س ش ص ض
ط ظ ع غ ف ک ق گ ل م ن ہ ھ ی

Table 6: Joiner alphabets in Urdu

Initial	میرا (*mera*) (mine)
Medial	خدمت (*Khidmat*) (service)
Final	آرام (*Aram*) (rest)

Table 7: Shapes of joiners in Urdu

آگجلادو (*aagjala do*)

Table 8: Example Sentence

Non-Joiners However, some letters have just one and final shape and they may join their preceding letters but do not connect with letters that are written after them and do not change their shape. These are called non-joiners. List of Non-Joiner alphabets is given here. Non-joiners can be written without adding space between them and it does not damage the word or sentence. For example, in the sentences (کاغذپرتصویربناؤ) (*kagaz par tasweer bnao*) (Make a picture on the paper), words are written without a single space between them. It can be seen that omitting space does not affect the sentence, because in these sentences each word is ending with a non-joiner. The sentences in Table 10 are written without a single space between the words.

اآدڈذرزڑژوءے

Table 9: Non-Joiner alphabets in Urdu

میزپراخبارپڑاہے (Newspaper is on the table)
مجھےکھانالادو (Bring me food)
بچےکوکھلونےدو (Give the baby toys)
کاغذپرتصویربناٶ (Draw a picture on paper)
اتوارکےبعدسوموارآتاہے (Monday comes after sunday)

Table 10: Space Standardization

It becomes challenging to tokenize such type of words on the basis or space factor. In English languages each word is separated by space but such type of words in Urdu does not need space to separate them. There is a need of standardization for space inclusion between Joiners and Non-Joiners. A standard could be defined that one should include space after each single word whether it ends on a Joiner, Non-Joiner or it is a compound word. For example the sentence with all the words ending on a non-joiner (کاغذپرتصویربناؤ) (*kagaz par tasweer bnao*) (Make a picture on the paper) and the compound word (شب‌وروز) (*sHab-o-roZ*) (day and night), should be standardized to be written as illustrated in Table 11.

کاغذ(space)پر(space)تصویر(space)بناٶ۔
شب(space)و(space)روز۔

Table 11: Non-Joiner alphabets in Urdu

3.3.3 Connected Stems

Sometimes two stems are joined together by omitting the space between them. Last word of the first stem joins the first word of second stem and both combines to make a single stem. For example two Urdu words (کون سا) (*kon-sa*) (which one) can also be written jointly as (کونسا)(konsa), (اس کا) (*us ka*) (his/her) can also be written as (اسکا) (*uska*). So it becomes difficult to tokenize such type of words because two words could be written as a single word without any space between them. There should be standardization about such words that all the words must be separated with a single space before and after them. For example it could be specified that writing the words (کون سا) like (کونسا) is incorrect and writing (کون سا) is correct only. Such type of standards could be defined to solve these issues.

3.4 Challenges in Urdu Stemming

Stemming is the process of extracting the root word from any given word (Riaz, 2007). Stemming is performed on both inflected and derived words. A lot of challenges arise while doing stemming in Urdu language. In this section we have discussed the challenges that are face in developing a stemmer for Urdu language.

3.4.1 Stemming the Infixes

In English language, inflectional of derivational words are created by adding affixes to start or end of the stem e.g. "Unavoidable" is created from the root word '"avoid". It has two affixes attached as "Un" is the prefix and "able" is the suffix. If we remove the prefix and suffix, we can get the root word. But Urdu language has a different case. The biggest problem in Urdu stemming is extracting stem from Infixes. Such words do not have any prefix or suffix attached with them rather they are modified through infixes. Some Urdu stems extracted from the infixes are given in Table 12.

It is observed that words having infixes follow certain patterns. Correct stem could be extracted from such words if patterns for such type of words (having infixes) are made (Abdul Jabbar and Khan, 2016). We have tried to list out all the possible word patterns based on length followed by infixes in Urdu language. Such words can be easily stemmed if rules are made following these patterns. These patterns are given in Table 13.

علوم *Aaloom* (knowledge)	علم *Elm* (knowledge)
وکلاء *wukla* (lawyers)	وکیل *wakeel* (lawyer)
وظائف *waZaif*(scholarships)	وظیفہ *waZeefa* (scholarship)
آداب *Aadab* (manners)	ادب *adab* (manner)
کتب *kutab* (books)	کتاب *kitab* (book)
فقراء *fuQraa* (beggers)	فقر *fqr* (hunger)
اشکال *asHkaal* (shapes)	شکل *sHakal* (shape)
اقوام *aQwam*(nations)	قوم *Qaom* (nation)
فوائد *fawaed* (benifits)	فائدہ *faedah* (benifit)
رفقاء *rufQaa* (partners)	رفیق *rafeeQ* (partner)

Table 12: Example Urdu Infixes

3.4.2 Ambiguous Affixes

Some words fall in the list of Affixes but are actually the part of the stem itself. If the stemmer chop these words considering them as affixes, the resultant word becomes meaningless. These types of words must be

Length 4		
1	فعال	(C_1+C_2+ا+C_4)
2	فعول	(C_1+C_2+و+C_4)
3	فعیل	(C_1+C_2+ی+C_4)
4	فاعل	(C_1+ا+C_3+C_4)
5	تفعل	(ت+C_2+C_3+C_4)
6	فعلا	(C_1+C_2+C_3+ا)
Length 5		
1	افعال	(ا+C_2+ا+C_4+C_5)
2	مفعول	(م+C_2+C_3+و+C_5)
3	مفاعل	(م+C_2+ا+C_4+C_5)
4	تفعیل	(ت+C_2+C_3+ی+C_5)
5	فعالت	(C_1+C_2+ا+C_4+ت)
6	فعولت	(C_1+C_2+و+C_4+ت)
7	فعیلت	(C_1+C_2+ی+C_4+ت)
8	فواعل	(C_1+و+ا+C_4+C_5)
Length 6		
1	افتعال	(ا+C_2+ت+C3+ا+C_6)
2	تفاعیل	(ت+C_2+ا+C4+ی+C_6)

Table 13: Infix Patterns in Urdu

identified as a part of the stem but not an Affix. The stemmer must be able to differentiate between real affixes and those that are part of the stems. There can be ambiguity in prefixes as well as suffixes. For example in the word (shop) (دوکان) -/*dookan*/, it appears that this word contain a suffix -/*an*/(ان), if it is removed then it produced a word -/doo-k/(دوک), which is a wrong word that gives no meaning. Similar is the case with ambiguous prefixes. For example in the word (flight) (پرواز) -/*parwaz*/, it appears that this word contain a prefix (پر) -/*par*/, if it is removed then it produced a word (واز) -/*waAz*/, which is a wrong word that gives no meaning.

Such type of words should be treated as an exceptional case (Abdul Jabbar and Khan, 2016). List of such words should be maintained in order to avoid these ambiguities.

3.4.3 Stemming Errors

While developing a stemmer, it is necessary to keep an eye at the details of what documents are being missed, or what documents are being incorrectly retrieved because of stemming errors. There are two types of Stemming errors over-stemming and Under-stemming.

Over-stemming happens when the algorithm removes too much of the suffix. It gives the words that shouldn't be grouped together by stemming, but are. For example three words (guest) (مہمان) -/*mehman*/, (adventurer) (مہم جو) -/*muhim-ju*/, (adventures) (مہمات -/*muhim-mat*/ gives an example of over stemming. These three words will be conflated to a common stem (adventure) (مہم) -/*muhim*/, removing the (ات) -/*aat*/, (جو) -/*ju*/ and (ان) -/*an*/ respectively, considering them as suffixes. That is a correct result in case of first two words but incorrect for third word because (مہمان) is a stem itself. It feels like it contains the stem (مہم) but actually it does not. Such type of stemming errors should be avoided. Over stemming can be avoided by setting a minimum size of the constraint of the derived stem (Abdul Jabbar and Khan, 2016).

Under-stemming happens when the stemmer leaves the suffix attached to the word. This refers to words that should be grouped together but aren't. For example, from word (anger) (ناراضگی -/*na-raZ-gi*/ the stemmer will remove the suffix (ی -/*yey*/ and the stem will remain (ناراضگ -/*na-raZ-g*/ and that is not a valid stem. The stemmer should remove the suffix (گی) -/*gi*/ to get the correct stem that is (angry) (ناراض -/*na-raZ*/. The problem of under stemming can be avoided by using techniques like partial-match algorithms (Abdul Jabbar and Khan, 2016).

3.4.4 Stemming the Plurals

Urdu language has two kinds of plurals: Unaltered Plural and Altered Plural.

Altered Plural (جمع مکسّر) -/*jamA-mukassar*/ is a form where the original stem and the balance are altered and the stem is changed. In Table 14 we have given some examples of altered plurals. Extracting stems from such kind of plurals is very difficult because whole form of the stem gets changed in it. That makes it difficult to get the stem from it. Such words in Urdu language are based on certain patterns. These words can be successfully stemmed if their patterns are correctly identified.

آداب (*Adaab*) (manners)	ادب (*adab*) (manner)
علماء (*ulmaa*) (scholars)	عالم (*Aalim*) (scholar)
خطوط (*kHatoot*) (letters)	خط (*kHatt*) (letter)
وجوہ (*wajooh*) (reasons)	وجہ (*wajah*) (reason)
اشکال (*shakal*) (shapes)	شکل (*shakal*) (shape)

Table 14: Examples of Altered Plurals

Unaltered Plural (جمع سالم) -/*jamA-saalim*/ is a plural form in which the original stem and the balance are not altered. Few examples of unaltered plurals are given in the Table 15. Let's take the word (حاضرین) -/*HaZir-een*/. The stem is (حاضر) -/*HaZir*/ that remains unaltered in the plural. Extracting stems from this form of plurals can be much easier. Stem remains unchanged in it and some extra words are attached to it. If this extra word is removed, stem can be extracted.

حاضرین (*HaZreen*) (presenters)	حاضر (*HaZir*) (present)
ثمرات (*samraat*) (results)	ثمر (*smar*) (result)
لڑکیاں (*laRkiyan*) (girls)	لڑکی (*laRki*) (girl)
کھڑکیاں (*kHiRkiyan*) (windows)	کھڑکی (*kHiRki*) (window)
ناظرین (*naZreen*) (viewers)	ناظر (*naZir*) (viewer)

Table 15: Example of unaltered plurals

Broken Plurals Extracting stems from broken plurals is also a challenging task in Urdu. There are certain words in Urdu language that have more than one possible plural of them. Such words are considered to be broken plurals of that word. For instance (rituals) - (رسمیں) (*rasmain*),(رسوم) (*rasoom*) (رسومات) (*rasoomat*) any of these can be used as the plural of the word (ritual) (رسم) (*rasm*). In order to stem such words one has to find out all the possible plurals of a word and to take care of all the possible inflections and derivations that a single word can have.

رسم (*rasm*) (ritual)	(rituals) (رسمیں, رسوم, رسومات)
وعدہ (*waEdah*) (promise)	(promises) (وعدے, وعدوں, وعود)
سجدہ (*sajdah*) (prostrate)	(prostrates) (سجدے, سجدوں, سجود)
وجہ (*wajah*) (reason)	(reasons) (وجوہات, وجوہ)
کتاب (*kitab*) (book)	(books) (کتابوں, کتب, کتابیں)

Table 16: Example of broken plurals

3.5 Word Sense Disambiguation Issues

Word Sense Disambiguation is the process of identifying which sense (meaning) of a word is used in a given sentence, when the word has multiple meanings.

3.5.1 Homonyms

Homonyms are the words having same spelling or pronunciation but different meanings and origins. In Urdu language, Homonyms are called (ذومعنی الفاظ). The Urdu word (بیت) (*bait*) holds two different meanings. It may give the meaning of “House” (گھر) or “Poetry” (شعر). Such type of words is context sensitive. In order to get the meaning of this type of word in a particular sentence, we need to know the context of that word. Meaning of that word depends on the position of that word in the sentence. This is problematic in Word Sense Disambiguation (WSD). Such words cause a number of problems for NLP applications like Machine translation, Text to speech and Information Retrieval. Serious orthographic errors may occur if context is not considered while translating such words. While doing information retrieval, in a word like (میں) machine doesn't know whether the word (میں) (*main*) means ”I” or it means ”in”. This type of words creates ambiguity in the text. Few more examples of Urdu Homonyms are given in the Table 17.

Word	Meaning 1	Meaning 2
عاجز (*Aajiz*)	down to earth	fed up
چشمہ (*cHashmah*)	fountain	glassess
اتفاق (*ittefaQ*)	incidentlly	unity
کان (*kaan*)	ear	mine
ملکہ (*malkah*)	expertise	queen

Table 17: Example of Urdu Homonyms

3.5.2 Homophones

Homophones are defined as: “Two or more words having the same pronunciation but different meanings, origins, or spelling”. So homophones are the words that give same sound but are different in writing and meaning. Such words sounds the same but spelled differently. In Urdu language, Homophones are called (متشابہ الفاظ) (*mutasHabah alfaaz*). While working on Speech to Text and vice versa, in Urdu language, homophone becomes big a challenge. Urdu letters that give same sound but are different in use. Examples of Homophones in Urdu are given in the Table 18. Urdu alphabets given in Table 19 posses same sound but are different in use. How to deal with them? When we are doing speech synthesis, it is difficult to differentiate between such types of words.

Word	Homophone
روضہ (*Aajiz*) (shrine)	روزہ (*Aajiz*) (fast)
عاری (*Aajiz*) (fed up)	آری (*Aajiz*) (saw)
آم (*Aajiz*) (mango)	عام (*Aajiz*) (common)
نذر (*Aajiz*) (offer)	نظر (*Aajiz*) (eyesight)
شعر (*Aajiz*) (verse)	شیر (*Aajiz*) (lion)

Table 18: Example of Urdu Homophones

Alphabet	Homophones
ا	آ, ع
س	ص, ث
ح	ہ, ھ
ت	ط
ک	ق
ز	ض, ظ, ذ
ژ	ی

Table 19: Example of Urdu Homophones

3.5.3 Diacritical Marks

There are some special characters in Urdu that lies above or below a letter, called Diacritical Marks (علاماتِ تلفظ). Diacritical marks aren't very common in Urdu writing. These are Zabar ◌َ, Zair ◌ِ, Paish ◌ُ, Madaa ◌ٓ, Shadd ◌ّ etc. Diacritical marks are not compulsory in Urdu. Mostly they are added only to help in pronunciation. Diacritical marks are not often used. These diacritics change the pronunciation and the meaning of the word and differentiate letters of similar shape with each other. If a didactical mark is added on a letter, it changes whole meaning of the word. For example the word (دور) can be used in two meanings by adding Paish ◌ُ or Zabar on (و) (*wao*). If we add Zabar ◌َ on (و) (*wao*) it becomes (دَور) (*daor*) which means ”Era”. If we write Paish ◌ُ on و (*wao*), it becomes (دُور) (*door*), which means ”Far away”. If diacriticals marks are not used on such words, it becomes difficult to interpret the meaning of that word. In the example below, same word (کھلا) is used in two different sentences in two different meanings by using diacritical marks. The word ((کھلا) with Paish ◌ُ on (ک) (*kaaf*) gives the meaning of ”Opening” and with Zair ◌ِ on (ک) (*kaaf*) gives the meaning of ‘Flowers bloom’:

کمرے کا دروازہ کھلا
kamre ka darwaZah kHula
باغ میں ایک پھول کھلا
baagH main aik pHool kHila

In this way, use of diacritical marks can change the meaning of whole sentence. Few more examples are given in the Table 20.

Word 1	Word 2
ہُوا (*huwa*) (happened)	ہَوا (*hawa*) (wind)
دُھن (*dhun*) (passion)	دَھن (*dhan*) (money)
تَیر (*tair*) (to swim)	تِیر (*teer*) (arrow)
دُم (*dum*) (tail)	دَم (*dam*) (breath)
سَر (*sar*) (head)	سُر (*sur*) (Tone)

Table 20: Example of Urdu Diacritics

3.6 Translation Issues

Urdu language proves to be a difficult language while translating it to any other language like English. The reason is its complex grammar and rich morphology and orthography. Following section describes the translation issues that are faced while translating Urdu text to English.

3.6.1 Translating Idioms

Idioms become extremely difficult to translate from Urdu to any other language like English. A word-for-word translation of Urdu idiom (ضرب الامثال) (*Zarb-ul-amsaal*) is often nonsense or changes the meaning of whole sentence. If we take an Urdu idiom: (آٹھ آٹھ آنسو رونا) (*AatH AaTh Aansu rona*) if we translate it to English word-for-word, it gives the translation: ”Eight tears to cry” and that is totally a wrong translation. This causes some serious orthographic issues. Some example Urdu idioms and their word for word English translation are given in the table below:

دانت کھٹے کرنا	(To sour teeth)
چار چاند لگانا	(Put four silver)
اینٹ سے اینٹ بجانا	(Brick by brick playing)
آ بیل مجھے مار	(Bulls come kill me)
بھینس کے آگے بین بجانا	(Buffalo ahead harp)

Table 21: Example of Urdu Idioms

3.6.2 Non-Equivalent words

While translating from Source Language (ماخذ زبان) (*mak-Haz zuban*) to Target Language (ہدف زبان) (*hadaf zuban*), Non-Equivalent words are those words that have no alternative in the target language. When we translate from Urdu to another language, if there is no equivalent of a word in the target language, then what translation would be used? Non-Equivalent words results in wrong translations. For example, when translating Urdu to English, a word (چائے پانی) (*chaey paani*) is translated as “Tea and Water.” But in Urdu language this compound word is actually used as the money and favors given to someone. Few more examples of the words having no alternate in English language are given in Table 22. Similarly the names given to relationships in Urdu language can never find an equivalent in English language. Few examples of the names of relationships in Urdu language having no alternate in English are given in Table 23.

رم جھم	(*rimjHim*)
ساون	(*sawan*)
مصالحہ	(*maSalah*)
حُقہ	(*huQah*)
دھمال	(*dHamal*)
گویا	(*goya*)

Table 22: Example of Non-Equivalent Words

پھوپھی	پھوپھا
تائی	تایا
خالہ	خالو
چاچی	چاچا
ممانی	ماموں

Table 23: Relationship Names in Urdu

3.6.3 Transliteration Issues

Urdu to Roman transliteration is difficult because there is no standardization on the spellings. While doing transliteration from Urdu to roman, different spellings are used by different people. There is more than one way of writing a particular word of Urdu in roman and all of them can be valid because there is no standardization. For example the word (میں) (*main*) can be translated as ”Me” or ”Main”. Some transliterated words of English are also used in Urdu. Few examples of different spellings of same word in roman are given in Table 24.

ہوں	hun, hon, ho
وہ	woh, wo, vo
مجھے	mujhy, mujhe, mujy
ہم	hum, ham, hm
میں	Main, me, mai

Table 24: Transliteration in Urdu

3.6.4 English Loan Words

Urdu is a language that keeps on evolving with time. Many English words are also included in today's modern Urdu language and are also being used by native speakers of Urdu language. People have converted these English words into Urdu according to their own understanding. For example the English word "Editor" and "Editors" are used in Urdu like (ایڈیٹر) (*ediTar*) and (ایڈیٹرز) (*ediTarz*) respectively. English plurals are also used in Urdu by adding the suffix (وں) (*aon*) with the English word converted to Urdu e.g. the word "Editors" is written as (ایڈیٹروں) (*ediTron*) when converted to Urdu language. Such loan words become difficult while stemming or translating in Urdu language. In Table 25 we have presented few examples English loan words in Urdu.

بینک	bank
پینسل	pencil
کمپیوٹر	computer
پروگرام	program
لائٹ	light
پولیس	police

Table 25: English Loan Words in Urdu

4 Conclusion

Urdu is a grammar enriched language. Many problems arise while performing any Natural Language Processing task on it. Current work briefly presented these problems and challenges. The overall goal of this work is to figure out all the morphological and orthographical challenges faced in Urdu language processing and to present the summary of these challenges. This study can help many new researchers that are trying to develop applications for Urdu language. Efforts are being made to solve the problems. Most of these problems identified in this work can be solved by making proper standards for Urdu language processing and a little more effort on computational matters can solve these problems. Problems faced by tokenization and Sentence Boundary Disambiguation (SBD) can be handled more effectively by using statistical methods instead of using rule based approaches. Issue like space inclusion and exclusion can be solved by standardizing the Urdu text writing in all disciplines. The paper provides a quick review of challenges that a researcher can face while working in Urdu Language Processing (ULP).

References

Abdul Jabbar, S. I. and Khan, U. G. (2016). A survey on urdu and urdu like language stemmers and stemming techniques. *Artificial Intelligence Rev.*

Ali Daud, W. K. (2016). Urdu language processing: a survey. *Artif Intell Rev.*

Hussain, S. (2007). Computational linguistics (cl) in pakistan: Issues and proposals. *Future Directions in Information Access (FDIA).*

Riaz, K. (2007). Challenges in urdu stemming (a progress report). *Future Directions in Information Access (FDIA).*

Sajjad Ahmad, Waqas Anwar, U. I. B. (2011). Challenges in developing a rule based urdu stemmer. *Proceedings of the 2nd Workshop on South and Southeast Asian Natural Language Processing (WSSANLP).*

Schmidt, R. L. (1999). Urdu: An essential grammar. *Lectures in the Department of East European and Oriental Studies.*

Waqas Anwar, Xuan W., X. L. W. (2006). A survey of automatic urdu language processing. *Proceedings of the Fifth International Conference on Machine Learning and Cybernetics.*

Zobia Rehman, Waqas Anwar, U. I. B. (2001). Challenges in urdu text tokenization and sentence boundary disambiguation. *Proceedings of the 2nd Workshop on South and Southeast Asian Natural Language Processing (WSSANLP), IJCNLP.*

Enhancing Semantic Role Labeling in Hindi and Urdu

Aishwary Gupta, Manish Shrivastava
IIIT Hyderabad, India
aishwary.gupta@research.iiit.ac.in, m.shrivastava@iiit.ac.in

Abstract

This paper presents a supervised semantic role labeler for Hindi which can be extended to Urdu as well. We propose a set of new features enriching the existing baseline system for these languages. We break the system into two subsequent tasks - Argument Identification and Argument Classification respectively. Our experiments show a reasonable improvement with respect to the current baseline for Hindi, mainly for the classification step. We also report significant improvements for Argument Identification task in Urdu. Finally, we create a new baseline for the Hindi using 5-fold cross-validation and we capture results excluding the null class and including the null class exclusively. We also extend the same work on Urdu and report the results.

Keywords: Semantic Role Labeling, PropBank, TreeBank, CBOW, Word2Vec

1. Introduction

In the last decade, there has been a lot of interest and a great amount of contribution towards semantic analysis of languages. There has been a significant amount of work done for major languages like English which included efforts in making semantically annotated data like PropBank. But only a little effort has been shown in Indian languages such as Hindi and Urdu. The Hindi PropBank and the Urdu PropBank were proposed just a few years back following which the first system, a semantic role labeler(SRL) for these languages was built. We saw a need for improvement in this domain and thus we introduce a new system with an additional set of features which makes a significant improvement in the classification of semantic roles for Hindi and extend the same work for Urdu. The major objective of SRL is to provide all sorts of information from a sentence in the form - who does what, where, to whom, where, when etc. In the PropBank, each sentence can be thought of as an event(s) which has participants - analogous to predicate having arguments. This labeling is done at phrase(chunk) level. The verb is the predicate and phrases/chunks related to it are its arguments labeled in categories such as Doer, Receiver, location, temporal etc. A SRL system has to therefore, label the arguments for each predicate of a sentence automatically.

Most of the previous works like (Pradhan et al., 2005), (Punyakanok et al., 2004), (Koomen et al., 2005) and (Anwar and Sharma, 2016) use a 2 step approach, i.e., first a chunk is identified whether it is an argument for a given predicate in the sentence or not. If yes, then it is classified at second step into the role labels. We use the same approach for reasons given in Section 3.1.

The applications of SRL can be seen in various research areas in Natural Language Processing(NLP). It can be attributed to the fact that semantic role labeling provides the meaning of a sentence at an abstract level. It can be seen in fields like information extraction (Christensen et al., 2010), question answering (Pizzato and Mollá, 2008) and machine translation (Liu and Gildea, 2010). Our paper is organized as follows:

Section 2 gives a brief description about the Hindi and the Urdu PropBank and how these were annotated above their respective TreeBanks. We also talk about the language resource we used for our task. Section 3 gives an idea about the related work done for this task. Section 4 shows our detailed approach and the system architecture. It also talks about the classifier we have used which helps us cut down the argument-identification task. In Section 5, We talk about the current best system as the baseline and then talk about the new features we have proposed. In section 6, we show how we conducted our experiments and the results for both languages. This also includes the comparison of our system with the existing system.

2. The Hindi Propbank and The Urdu Propbank

The Hindi Treebank and The Urdu Treebank were added with a layer of semantic annotation to give rise to The Hindi Propbank (Vaidya et al., 2011) and the Urdu PropBank (Anwar et al., 2016) respectively. These are part of the Hindi-Urdu PropBank Project which is a multi-dimensional and multi-layered resource creation effort for the Hindi and the Urdu language (Bhatt et al., 2009). Unlike PropBanks in most of the other languages, these PropBanks are annotated on top of the corresponding TreeBanks which have a dependency structure. The Treebank already having the dependency annotation, now including lexical semantic information at chunk level forms the PropBank corpus. PropBanks of both the languages include dependency relations at the chunking level which help construct the sentence dependency tree, morphological information for each word, part-of-speech/syntactic category at chunk as well as token level. The PropBanks as well, similar to the TreeBanks, are represented in the Shakti Standard Format (Bharati et al., 2007). The sentences are distributed in various documents. Each document has 15-20 sentences where each sentence is broken down into chunks and each chunk is broken down at token level.

Propbank labels‘ (or semantic role labels) annotation was made easy by dependency relations - also called as karaka relations (Vaidya et al., 2011) (described later) because there is a close syntactic-semantic relation in them. In the PropBank, semantic roles are defined for each verb which means that a fixed set of roles are specified for each verb and a distinct label is assigned to each role. These are labeled in different ways in various PropBank annotations. For Hindi, the core arguments are the numbered arguments which are labeled as ARG# where #-{0,1,2,3}. For example, the verb bawA(to tell), has four such numbered arguments: ARG0: person telling, ARG1: thing told, ARG2: hearer there is no ARG3 for this verb. An important point to be noted here - an argument marked with the same number, say. ARG0, may not share any semantic similarities for different verbs. Further, each verb can have a different set of these arguments depending on the use of the verb in a sentence. This is handled by providing different frameset/sense to each verb which means that the annotation also has the information of which way the verb is being used in a sentence. Example- the same verb bawA - has another meaning which is - to mention/describe and hence has a slight difference in its set of arguments, namely, ARG0: the one mentioning or describing A as B, ARG1: the thing A that is described as B, ARG2-ATR: the description B that is used to describe A. The Hindi PropBank has distributed ARG2 into 4 more labels namely - ARG2-ATR(attribute), ARG2-LOC(location), ARG2-GOL(goal), ARG2-SOU(source). There are also certain other modifier labels denoted as ARGM* which are not specific to any verb and can be shared by any verb. The Hindi PropBank has 24 distinct labels and the Urdu PropBank has 20 distinct labels with number of modifiers being 4 less than those in Hindi.

2.1. Dataset

As reported earlier (Anwar and Sharma, 2016), for Hindi PropBank they took around 100,000 tokens as training data and 20,000 as test data and for Urdu they took 130,000 tokens as training data and 30,000 as test data. We have used exactly same data for Phase 1(Section 6) of our experiments.

3. Related Work

According to the best of our knowledge, only work done on automatic semantic role labeling for Indian Languages PropBank, i.e., Hindi PropBank and Urdu PropBank was seen last year (Anwar and Sharma, 2016). Other than this, on English PropBank, plenty of work has been done. One of the earliest work on SRL on English PropBank(2001) was done by Gildea and Jurafsky (2002). Xue and Palmer (2004) showed that full exploitation of the syntactic tree was needed in the earlier stages to improve the results for semantic role labeling. Towards the recent years, Roth and Woodsend(2014) have shown that vector representation of predicate, arguments and also composition of words leads to improve semantic role labeling.

Since a single system has been made for semantic role labeling for Indian Languages. we take it as the best model and compare our system with them.

4. Semantic Role labeler

Depending on the type of information one wants to learn automatically, there are various ways to construct the semantic role tagging task resting on the annotation of the PropBank of that language. Following the previous work (Anwar and Sharma, 2016) for comparison purposes, we ignore the frameset/word-sense information for now. Therefore we will predict the numbered core arguments ARG[0-3], ARG2 x secondary tags and all ARGM* tags, for each predicate in a sentence. There are also some phrases/chunks in a sentence that are not semantic arguments for predicate in concern and we will label such chunks as NULL. Semantic role labeling can thus be comprehended as a 1 of N classification task but so is not the case. Let us look why in the next section.

4.1. Selecting Approach

As shown in the previous work on Indian Languages (Anwar and Sharma, 2016), direct classification of roles without filtering NULL arguments gave very poor results as compared to the two step approach. In one of the earliest work (Xue and Palmer, 2004), it is observed that for a given predicate, many chunks in the syntactic/dependency tree don't act as its semantic argument. So, the null class count overwhelms the argument count for the given predicate and classifiers will not be efficient in predicting the right argument or classifying them. Also, the features required for checking whether a chunk is an argument or not can be different from the features used to classify roles. Another reason for using this architecture is that it saves a lot of training time for the classifier in the second step. Hence, we follow the 2 step approach, i.e., first identifying the null labels and then classifying the rest. Therefore, we first train a binary classifier to label each chunk as a semantic argument or not. For hindi, the reduces the training data by 51% and for Urdu it is reduced by 81%. Some of the NULL arguments also go to the next step (10% for Hindi and less than 1% for Urdu). Also, some of the non-NULL arguments are filtered out in the first step. Second, we train a multi-class classifier to label the chunks in all classes including the NULL class.

4.2. System Architecture

We do semantic role labeling at a phrase/chunk level. We can break our approach in three major steps along with null data chunk removal as the 0th step.

Step 0: From the dataset we chose, we simply do not take the sentences which have no semantic annotation, i.e., we remove the sentences not having the argument labels and information about the verbs("pbrole annotation") and their frames. If no information is present, we remove the sentence.

Step 1: We run a binary classifier to classify the constituents as Arguments or Non-Arguments(NULL).

Step 2: We run a multi-category classifier to classify the constituents that are labeled as arguments into one of the classes plus NULL.

For the 2nd and final step, we used a Support Vector Machine(SVM) Classifier from the sci-kit library(in Python). Their SVM is a multi-class classifier which learns unique boundaries for each class by taking one vs rest approach for training every class. The classifier‘s soft boundary can be tuned to maximize results till there is not over-fitting. We also tried our hands using a simple 2 layered neural network having the 1st layer equal to the number of features(intuition based) and the last layer equal to the number of classes. We see the outputs from it were also similar which tells us that in our case it largely depends on the features what we give to a machine irrespective of the classifiers. Let us take a look at the features used in previous work and the advancements that could be done.

5. Features

First, we go through the some of the features and techniques used in previous works. We only take the features from the previous baseline that we have used in our system also. We then show the features introduced by us to improve the performance of the system from the current baseline.

5.1. Baseline Features

We take features from the previous system and consider them as baseline for us.

Predicate - predicate word is taken as it is.
Head-word - Head of the chunk/phrase according to syntactic-relations.
Head-word POS - Its Part-Of-Speech category.
Chunk Type - syntactic category (NP, CCP, VGF etc. of the chunk)
Dependency/karaka relation - syntactic relations between chunks.

We look at the use of the above features for both Argument Identification and Argument Classification tasks. The predicate alone cannot tell us any information about identification or labeling but when it is used with other feature such as the head word and head-word's POS, then only it makes sense whether this head word‘s chunk belongs to a label or not. The head word is an important feature as some of the chunk heads are more likely to be certain arguments for a predicate. This also accounts for the use of predicate as a feature. We use the head-word POS tag along the above because of similar reasons. When used with the predicate, the phrase/chunk tag is useful for identification task because for a predicate, a certain tagged chunk will be more probable to be an argument or a non-argument. The use of karaka relation, a property from the syntactic dependency tree was shown to be one of the best features in this task (Anwar and Sharma, 2016) for Hindi and Urdu. Also, as an inspiration from “Analysis of the Hindi Proposition Bank using Dependency Structure” (Vaidya et al., 2011), we incorporate this feature because of the mappings in their paper show that there is a good interrelationship between the syntactic and semantic predicate-argument relations in a sentence. On account of similar reasons, we also use these features for the argument classification task.

5.2. New Features

After analyzing the PropBanks of both the languages, we came up with certain new features for which we had an intuition that they will contribute significantly towards this task. These are discussed below:

5.2.1. Argument Identification

The following features are added for Argument Classification task as well.

Predicate(verb)'s root form and suffix features - Using the predicate word directly as a feature increases the number of unique instances for the same. To tackle this, we use break the word into its root/stemmed form plus its suffix. This highly reduces the number of distinct verbs for our system as many words fall into the same root category which in-turn gives a boost to our results. For example, in English the predicate ‘play‘ can be present in a sentence as ‘playing‘ or ‘played‘ but both fall under the same predicate. Therefore we take the root ‘play‘ and the suffix ‘ing‘ or ‘ed‘ separately.

Head-Word embedded as a vector - This is the most important feature which contributes in lifting up our results. The reasons to switch over to a new representation are quite similar to the reason for predicate word. In this case, the number of distinct words are a lot more(4300 for Hindi and 4100 for Urdu) because of the richness of the languages used. In a language, the number of verbs(in their root form) are indicatively smaller than the nouns or pronouns. Majority of the chunks we label are of the syntactic category-NP and hence the heads will be nouns or pronouns in a good amount which increases the complexity of this feature. Referring Gensim‘s Word2Vec, we experimented using their Continuous Bag Of Words approach to create a vector representation of head-word which is of size 30. This highly reduced the size of this feature and improved quality of our results.

Path - It has been shown in many of the previous works that path between the chunk and the predicate has been an important feature in argument identification if not classification. We call for this feature because in Hindi, certain path configurations are more likely to be the arguments to a verb than others. For example, a chunk with path NP↑VGF is more likely to be an argument and a chunk with path JJP↑VGF.
Along with this, we also use dependency path from the intuition that it may further help in classifying the chunks into argument labels according to their syntactic dependence in tree.

Parent and Grandparent‘s syntactic category - Going through the data and looking at tree structures of the sentences, it is evident that in a good number of cases, either the parent or the grandparent is seen as the predicate for a chunk. Along with the syntactic category we also use the parent-grandparent dependency relation to support the classification.

5.2.2. Argument Classification

The following features were only added for this task.

Chunk‘s Vibhakti/Post-positional - ‘Vibhakti‘ is a Sanskrit term which is used for post-positions and suffix in Indian languages. In case of Hindi which uses post-positions instead of prepositions as in English, the post-position similarly provides a good discrimination in selecting the semantic labels.

Other than these we also tried using speech and voice of the predicate chunk. We have not included it in the final system because the results declined adding them to the baseline. Also, there are some chunks in the data which belong to null syntactic categories (Begum et al., 2008). These are namely NULL__NP, NULL__VGF, NULL__CCP. While extracting the features and using them for training, we also appended their non-null categories for every feature where chunk‘s syntactic category is needed. For example, a NULL__NP chunk is also given the category NP at training time. In the next section, we show the results of these features and compare our system with the existing work.

6. Results and Experiments

We performed our experiments in two phases. The first phase includes experiments on the same train data and test data for both Hindi and Urdu as used in the earlier work (Anwar and Sharma, 2016). The second phase was to make the results more generic over the data and hence we did a 5-fold Cross validation making the train to test ratio as 80% train to 20% test.

Phase 1-
Argument Identification. The results for Hindi and Urdu are shown in table 1 and table 2 respectively. For this step, we trained a binary SVM classifier. We did experiments tuning the hyper-parameters of the classifier and finally got the best results by using - Penalty/Regularization parameter, C = 100.0 and we used a ‘rbf‘ Kernel with kernel coefficient, γ = 0.0005 for Hindi. For Urdu we used similar tuning with C = 80.0 and γ = 0.0006.

System	Precision	Recall	F-Score
Previous Baseline	88	87	87
This work	91.41	90.49	90.94

Table 1: Argument-Identification results for Hindi

System	Precision	Recall	F-Score
Previous Baseline	78	79	78
This work	92.05	91.49	91.76

Table 2: Argument-Identification results for Urdu

Argument Classification. We begin by building the results on baseline features in Hindi shown in table 3. In the next step, we conduct experiments for each of the new feature we propose. This helps us to see the gain and importance of individual feature in the system.
To convert our head-word to vector representation, we used Gensim‘s Word2Vec tool which is a python library having the Continuous Bag of Words(CBOW) (Mikolov et al., 2013) approach. To train that model on our language, for both Hindi and Urdu we used raw sentences from their corresponding Dependency TreeBanks. For both languages, we used around 200,000 tokens to train the Word2Vec model and used that model for our feature conversion. Table 4 and Table 5 shows the comparison of Argument-Classification(including NULL class) for Hindi and Urdu respectively.

Feature	Precision	Recall	F-Score
Baseline	56.04	49.55	52.59
+Predicate’s Root and Morph	60.29	52.88	56.39
+Head-word Vector	61.97	62.12	62.04
+Path	61.15	58.28	59.68
+Parent POS	59.15	55.56	57.29
+Grand-Parent POS	58.33	55.07	56.65
+Vibhakti POS	60.41	59.93	60.16
+Speech and Voice POS	55.32	49.47	52.23

Table 3: Argument-Classification Feature-Wise results for Hindi

System	Precision	Recall	F-Score
Previous Baseline	58	42	49
This work	65.01	66.62	65.80

Table 4: Argument-Classification results for Hindi

System	Precision	Recall	F-Score
Previous Baseline	87	85	86
This work	86.72	86.37	86.54

Table 5: Argument-Classification results for Urdu

Phase 2- In phase 1, the data was split in train and test as shown in Section 2 which is- For Hindi, 100,000 tokens were taken as training data and 20,000 tokens as test data which is a 83.33% train to 16.67% test data; For Urdu, 130,000 tokens as train and 30,000 tokens as test were chosen which is 81.25% train to 18.75% test data. Instead, we came up with a more generic approach which was to split the total data(train+test) used in both Hindi and Urdu to a 80-20 split which can provide for a 5-fold cross-validation. Hence, we present our results averaged after cross-validation as our final results for Hindi and Urdu in table 6 and table 7 respectively. The **Argument-Classification*** results are excluding the NULL class‘s contribution in the result.

Task	Precision	Recall	F-Score
Argument-Identification	91.08	89.93	90.50
Argument-Classification	64.26	65.90	65.06
Argument-Classification*	69.12	73.16	71.12

Table 6: 5-fold Cross-Validation for Hindi

The previous work did not report the results excluding NULL class making it difficult for comparison purposes since the results would vary dependent on how much NULL

Task	Precision	Recall	F-Score
Argument-Identification	93.79	93.43	93.60
Argument-Classification	84.71	84.89	84.80
Argument-Classification*	85.37	86.27	85.81

Table 7: 5-fold Cross-Validation for Urdu

arguments are present at the classification step. In other way it is dependent on the performance of the Identification task. All experiments for classification step were carried out using SVM classifier with hyper-parameters tuned as - C = 70.0 and we used a 'rbf' Kernel with kernel coefficient, γ = 0.0005 for Hindi. For Urdu we used similar tuning with C = 300.0 and γ = 0.0006.

7. Future Work

This paper does not focus on handling cases where multiple arguments of the same predicate are assigned the same role which is theoretically not possible. To avoid this, we need to use some re-ranking method to assign the best possible set of arguments for a given predicate such that the likelihood of our output is maximized. This can be handled using Integer Linear Programming (Punyakanok et al., 2004)

8. Bibliographical References

Anwar, M. and Sharma, D. (2016). Towards building semantic role labeler for indian languages. In Nicoletta Calzolari (Conference Chair), et al., editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, may. European Language Resources Association (ELRA).

Anwar, M., Bhat, R. A., Sharma, D., Vaidya, A., Palmer, M., and Khan, T. A. (2016). A proposition bank of urdu. In Nicoletta Calzolari (Conference Chair), et al., editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, may. European Language Resources Association (ELRA).

Begum, R., Husain, S., Dhwaj, A., Sharma, D. M., Bai, L., and Sangal, R. (2008). Dependency annotation scheme for indian languages. In *IJCNLP*, pages 721–726.

Bharati, A., Sangal, R., and Sharma, D. M. (2007). Ssf: Shakti standard format guide.

Bhatt, R., Narasimhan, B., Palmer, M., Rambow, O., Sharma, D. M., and Xia, F. (2009). A multi-representational and multi-layered treebank for hindi/urdu. In *Proceedings of the Third Linguistic Annotation Workshop*, ACL-IJCNLP '09, pages 186–189, Stroudsburg, PA, USA. Association for Computational Linguistics.

Christensen, J., Soderland, S., Etzioni, O., et al. (2010). Semantic role labeling for open information extraction. In *Proceedings of the NAACL HLT 2010 First International Workshop on Formalisms and Methodology for Learning by Reading*, pages 52–60. Association for Computational Linguistics.

Koomen, P., Punyakanok, V., Roth, D., and Yih, W.-t. (2005). Generalized inference with multiple semantic role labeling systems. In *Proceedings of the Ninth Conference on Computational Natural Language Learning*, pages 181–184. Association for Computational Linguistics.

Liu, D. and Gildea, D. (2010). Semantic role features for machine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 716–724. Association for Computational Linguistics.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Pizzato, L. A. and Mollá, D. (2008). Indexing on semantic roles for question answering. In *Coling 2008: Proceedings of the 2nd workshop on Information Retrieval for Question Answering*, pages 74–81. Association for Computational Linguistics.

Pradhan, S., Hacioglu, K., Krugler, V., Ward, W., Martin, J. H., and Jurafsky, D. (2005). Support vector learning for semantic argument classification. *Machine Learning*, 60(1):11–39.

Punyakanok, V., Roth, D., Yih, W.-t., and Zimak, D. (2004). Semantic role labeling via generalized inference over classifiers. Technical report, ILLINOIS UNIV AT URBANA-CHAMPAIGN DEPT OF COMPUTER SCIENCE.

Vaidya, A., Choi, J. D., Palmer, M., and Narasimhan, B. (2011). Analysis of the hindi proposition bank using dependency structure. In *Proceedings of the 5th Linguistic Annotation Workshop*, pages 21–29. Association for Computational Linguistics.

Xue, N. and Palmer, M. (2004). Calibrating features for semantic role labeling. In *EMNLP*, pages 88–94.

Preparing Bengali-English Code-Mixed Corpus for Sentiment Analysis of Indian Languages

Soumil Mandal[1], Sainik Kumar Mahata[2], Dipankar Das[3]

[1]Department of Computer Science & Engineering, SRM University, Chennai
[2,3]Department of Computer Science & Engineering, Jadavpur University, Kolkata
{soumil.mandal, sainik.mahata, dipankar.dipnil2005}@gmail.com

Abstract

Analysis of informative contents and sentiments of social users has been attempted quite intensively in the recent past. Most of the systems are usable only for monolingual data and fails or gives poor results when used on data with code-mixing property. To gather attention and encourage researchers to work on this crisis, we prepared gold standard Bengali-English code-mixed data with language and polarity tag for sentiment analysis purposes. In this paper, we discuss the systems we prepared to collect and filter raw Twitter data. In order to reduce manual work while annotation, hybrid systems combining rule based and supervised models were developed for both language and sentiment tagging. The final corpus was annotated by a group of annotators following a few guidelines. The gold standard corpus thus obtained has impressive inter-annotator agreement obtained in terms of Kappa values. Various metrics like Code-Mixed Index (CMI), Code-Mixed Factor (CF) along with various aspects (language and emotion) also qualitatively polled the code-mixed and sentiment properties of the corpus.

Keywords: code-mixed, sentiment classification, language tagging, Twitter data, social media analysis

1. Introduction

India has a linguistically diverse and vast diaspora due to its long history of contact with foreigners. English, one of those borrowed languages, became an integral part of the Indian education system and has been recognized as one of the official languages as well, thus giving rise to a population where bilingualism is very common. This kind of language diversity coupled with various dialects instigates frequent code-mixing in India. This phenomenon has become even more transparent with the rise of social networking sites like Twitter and Facebook and also instant messaging services like WhatsApp etc. The writing style in such media indicates phonetic typing transliterated in Roman, generally mixed with English words through code-mixing and also Anglicism. Three facts are involved in this sort of code-mixing cases, 1. lack of knowledge in using appropriate native words, 2. typing convenience and 3. popularity of Roman script to cater to a large set of audience.

Social networking services has been gaining popularity very rapidly since their first appearance and has led to an exponential growth of minable data which is rich and informative. In developing countries where majority of the population are bilinguals, in social media data, we frequent observe a unique trend in typing where two or more languages are mixed for expression known as code-mixing. It is also observed that such code-mixed data are growing rapidly in WWW because multilingual users in social networks frequently share their sentiments and thus it becomes an important task to mine and analyze such data for gathering crucial informatics related to sentiment too. However, the complexity involved in mixing of multiple rules of grammars, scripts, use of transliteration in such code-mixed data possesses a big challenge for NLP tasks. Thus, it becomes an ever so important task to solve this problem since a huge chunk of the data on social media possesses this property and will be of great use if mined.

It has to be mentioned that the conventional methods devised for a single language inevitably fail or give poor results in such cases. Thus to bring more attention of researchers towards this important and challenging aspect, we developed code-mixed corpora for sentiment analysis in Indian languages. India is country with 255 million [1] multilingual speakers and one of our goals in this was to challenge the participants and researchers into building advanced and robust systems for sentiment analysis of such code-mixed data. In the present article, we describe the systems and strategies used for making the Bengali-English code-mixed resources. Bengali is an Indo-Aryan language of India where 8.10% of the total population are the first language speakers and is also the official language of Bangladesh. The original script in which Bengali is written by locals is the Eastern Nagari Script [2]. Majority of our collected data is from Twitter. The reasons why Twitter is an ideal source for collection of such data has been explained by (Pak and Paroubek, 2010). The contributions of our paper are as follows:

1. A method for collecting code-mixed data using filtering techniques to assure quality and reduce manual effort.
2. A fast and reliable language identification algorithm (accuracy = 81%) for code-mixed data with known target languages.
3. A sentiment classification system for code-mixed data using a hybrid system (accuracy = 80.97%) combining rule based and supervised models.
4. Gold standard Bengali-English code-mixed data with language and polarity tags.

[1] http://rajbhasha.nic.in/UI/pagecontent.aspx?pc=MzU=
[2] https://www.omniglot.com/writing/bengali.htm

5. Several useful polarity tagged lexicons like phrasal lexicon of length 1200, uni-gram lexicon of length 3000 consisting of phonetically transliterated Bengali words, English acronyms commonly used on social media and a list commonly used emoticons.
6. Also, a seed list of length 1500 for querying Twitter API for retrieving Bengali-English code-mixed data.

2. Related Work

Several automated systems for Twitter data collection have been made before for corpus collection targeting different aspects but none with the aim to collect code-mixed data as far as our knowledge. On the other hand, various language tagging models have been made recently for code-mixed data and quite a few where a common script has been used for both the languages and one of them is phonetically translated. Among these one of the most relevant works is by (Das and Gambäck, 2014). Here they demonstrated a system which uses modified character n-gram with weights combined with a lexicon based approach, minimum edit distance as well as context info. (Barman et al., 2014) used a hybrid system by combining a lexicon based module with supervised classifiers like SVM, CRF and decision trees. Some of them have also been made as a sub-part for a part-of-speech tagging system like the one by (Vyas et al., 2014). For sentiment analysis on code-mixed, binary polarity classification has been tried using different classes of supervised models by (Ghosh et al., 2017b) and for ternary polarity by (Ghosh et al., 2017a) and (Sharma et al., 2015). A comparative study of classifiers trained on different code-mixed features was done by (Mandal and Das, 2018). Sophisticated methods using sub-word LSTM for learning sentiments in noisy code-mixed data has been tested as well by (Joshi et al., 2016).

3. Code-mixed Corpus Development

Corpus collection was done in two steps by collecting raw data from Twitter followed by filtering and cleaning code-mixed data from raw data.

3.1. Raw Twitter Data Collection

Our primary aim was to collect quality Bengali-English code-mixed data. However, we observed several instances of phonetically transliterated Bengali utterances (written in Roman script) that do not convey the code-mixed property (Muysken, 2000). We were also eager to collect intra-sentential i.e code-switched data instead of inter-sentential since the former is much more common on social media and is relatively more challenging for polarity classification as compared to the latter. For collecting Twitter data, we used the public streaming Twitter API via the Twitter4j [3] using keywords for querying. The initial keyword list was prepared by considering commonly used positive and negative Bengali words (e.g., bhalo, kharap, baje) and their polarities were validated using Bengali SentiWordNet (Das and Bandyopadhyay, 2010). We collected a total of 600 code-mixed sentences manually from the initial search output. In order to overcome the saturation problem of the retrieved data with respect to a few query words, we made a validated Bengali keyword list of 1500 unique query words from 600 sentences in decreasing order of frequency.

3.2. Data Filtering & Cleaning

The collected raw Twitter data contained noise, mostly contributed by words from other languages than the required pair, partially or fully (e.g. *bahar* which is a commonly used Hindi word meaning "outside"), words or full texts not in Roman script, etc. Thus, it was very important to build and apply a filtering module for retaining relatively better quality data in order to reduce manual efforts. Moreover, in order to avoid the problem of duplicacy due to short interval of querying, we have considered two parameters for devising our filtration strategy. The first parameter is α which denotes the minimum number of Bengali tokens with respect to our seed list whereas β refers to the minimum length of a tweet. It was observed that, the coverage of the top frequent keywords from the seed list helped us to filter majority of our code-mixed instances from the raw data if we vary the values of α only in the range of 1 to 3 and β in between 4 to 6. However, in order to filter more code-mixed instances for fulfilling our requirement, we had to increase the value of α up to 5 and the β up to 8 to maintain the code-mixed property in our filtered tweets. The total amount of raw tweets collected was around 89k and the our filtering system filtered out about 10k tweets from it. The statistics are shown in the Table 1. Here N denotes the information of n^{th} settings using which the Twitter API was queried, filtered data denotes the number of data remaining after removal.

N	α	β	Keywords Spent	Filtered Data
1	2	4	150	3800
2	2	5	250	2500
3	3	6	300	1800
4	4	7	350	1500
5	5	8	450	900
sum			1500	$\approx$ 10500

Table 1: Filtering statistics with respect to α and β .

During the cleaning process, spams, incomplete tweets, ones with conflicting sentiments were removed manually. Sarcastic tweets were not removed as it has become a very common tool for expression in the 21st century, especially on social media and thus it is important to classify them properly using more advanced techniques. URLs and Hashtags were kept as well as they too are important for sentiment analysis [4] (e.g. visiting the URL for analysis). We wanted to keep the data as untouched as possible to urge the future researchers to build highly robust systems which can be directly used on social media contents without much modification. Table 2 show the retrieved, filtered and used code-mixed data counts. It can be seen

[3] http://twitter4j.org/en/

[4] https://open.blockspring.com/bs/sentiment-analysis-from-url-with-alchemyapi

that our filtering system filtered out quite a lot of data and retained only about 11.79%.

Type	R	Count
Retrieved Tweets (RT)	≈	89000
Filtered Tweets (FT)	≈	10500
Code-Mixed Tweets (CT)	≈	5000

Table 2: Tweets retrieved statistics.

Some examples from our collected data after filtering are given below (underlined - EN, normal - BN) -

1. Thik fairy tale er ending tar moton amra shobaio happily ever after thakte lagilam. (Trans: *Just like a fairy tale ending we also lived happily ever after.*)
2. Script ta khub tiring chilo amar mote, aro onek better hote parto. (Trans: *The script was very tiring according to me, could have been much better.*)

4. Annotation

In order to annotate the language and sentiment tags to the filtered and cleaned tweets, we developed a system that help in basic annotation. One of the motivations of our annotation task was to reduce the manual tagging effort as we had to deal with huge amount of tweets ~10K. Therefore, in order to cope up with the problems of manual annotations, we planned to build two basic annotation systems, one is for language tagging and another is for sentiment tagging. Both of the annotation systems are described in subsection 4.1.. Finally, the outputs of these systems were evaluated by two sets of annotators, one set (A) consisted of a single annotator from Computer Science background with Bengali as mother tongue, where as the second set (B) consisted of five experts and the final evaluation was done by them. In order to handle the confusion cases, an annotation guideline as discussed in subsection 4.2. was provided to the annotators prior tagging.

4.1. System based Annotation

Out of 10k filtered tweets given by the system, we manually selected a collection of 5k tweets (as all filtered tweets were not code-mixed) and then we fed it the language tagging and sentiment tagging systems.

4.1.1. Language Tagging System

For language tagging, we used a two-step modular approach by combining lexicon based module (LBM) along with a supervised learning module (SLM).

LBM: As our target was simple, that is only to tag Bengali (BN) or English (EN) at word level, we tried to develop a relatively simple system. All the other unknown words are tagged as UN. The resources used to build the language tagging system are -

1. A list of Bengali words of size 3000 was prepared from the code-mixed data used in (Mandal and Das, 2018). Same words with different phonetic transliterations (e.g. *bhalo* and *balo*) both meaning good were also kept in the list.
2. *English Words (EW)* - A list containing 466k English words [5] was collected from online open sources.
3. *Suffix List (SL)* & *Acronym List (AL)* - An English suffix list [6] (e.g. *ing*, *ism*, *ious*) and an English acronym list [7] (e.g. *bbl*-be back later, *omg*- oh my god) was collected.
4. *N-Grams* - Bi-grams and tri-grams dictionary at character level was prepared from the above mentioned Bengali (BW) and English (EW) word lists, where keys were the n-grams and the respective values were frequency.

SLM: A supervised language tagger was developed by training the Linear Support Vector Machine (LSVC) implemented using scikit learn on two features which were character n-grams (n:2,3) as described in LBM features. For training, Bengali word list and list of most common English words [8] were used. The langauge tagging algorithm first searches the target token into our lexicons and if found, the appropriate tag is given. If not found, the supervised tagger is used to output the tag of that target token. The system was tested on ICON 2016 [9] POS tagging contest data and achieved a score of 86.24%.

4.1.2. Sentiment Tagging System

We used a hybrid system for sentiment classification. Similar to language tagging system, the sentiment tagging system also checks whether a tweet sentence is positive / negative / neutral using rule based method and if it fails, the supervised classifier is employed to produce the output sentiment tag. The resources which were prepared and used in the rule based were also used in supervised method as features.

Rule Based Method - For our rule based checking, three rules that were used to identify the sentiment of a tweet are as follows -

1. *Feeling (FLNG)* - A regular expression was used to extract the word that follows ’− feeling’ which is commonly used to express how the author feels. As such instances were self-tagged by the authors, there is no chance of ambiguity with respect to sentiment tagging. These tags are used since the stand alone texts may send different emotional signals or the author might simply be trying to convey his emotions directly.
2. *Hashtag (HT)* - Hashtags which used camel-casing or underscore separation were split and matched with lexicons and n-grams.
3. *Emoticon (EMO)* - Emoticons have a very strong impact on sentence level sentiment. We have used both

[5] https://github.com/dwyl/english-words

[6] https://www.learnthat.org/pages/view/suffix.html

[7] http://www.muller-godschalk.com/acronyms.html

[8] http://www.ef.com/english-resources/english-vocabulary/top-3000- words/

[9] http://ltrc.iiit.ac.in/icon2016/

Unicode and Icon representations of positive and negative emoticons for our experiments. Emoticon scoring has been experimented in three ways, e.g. higher frequency, greater index and average index. The second method which is based on the theory that the emoticon with the greatest index has the greatest influence on the tweet sentiment showed the best results.

Supervised Method - We have experimented with several supervised classifiers. In the Naïve Bayes (NB) family, we have used Gaussian Naïve Bayes (GNB), Bernoulli Naïve Bayes (BNB) and Multinomial Naïve Bayes (MNB). The Linear Models (LM) we have tested with are Linear Regression (LRC) and Stochastic Gradient Descent (SGDC). The scikit-learn [10] implementations of the models were used. The features used for supervised methods are as follows -

1. *Word N-Grams (WN)* - Word level uni-grams, bi-grams and trigrams were adopted as features. Each of the n-grams was sorted according to frequency in non-increasing order and the top 2000 n-grams were selected for training.
2. *Negation (NEGA)* - Negation in a message always reverses its sentiment orientation. If the number of negating words is odd, the polarity is reversed otherwise the calculated polarity is retained. Therefore, we collected a total of 25 English and 130 Bengali unique negation words.
3. *Tagged Words (TGW)* - We have also collected 1198 positive and 1802 negative Bengali uni-grams from an external code-mixed data available in (Mandal and Das, 2018). We combined them with English positive and negative words collected from NRC Emotion Lexicon and SOCAL lexicon to build a lexicon containing positive uni-grams (POSU) and negative uni-grams (NEGU).
4. *Tagged Phrases (TGP)* - In addition to words, we made a phrasal lexicon of length 1200 by extracting phrases ($\geq$ 1 from each sentence) from the code-mixed data described in (Mandal and Das, 2018). Such phrases are responsible to convey sentiment at the sentence level. For example, *boshe dekha jaye na* (trans: can't sit and watch), *onekei couldn't sleep* (trans: many couldn't sleep), etc. In case of tagged phrases, four scenarios were tested, *perfect match* - the phrase present in the sentence is identical to the tagged phrase, *sparse match* - all the unigrams of the tagged phrase are present in the sentence but not in the same order, *partial match* - a bigram from the tagged phrase (if $|\text{phrase}| \geq 2$) is present in the sentence in exact order (a bigram unit of stopwords is not considered) and finally, *no match* - none of the uni-grams is matched or the matched uni-gram is a stop-word.
5. *Tagged Acronyms (TA)* - Commonly used abbreviations on social networking sites were collected and polarity tagged as either positive or negative.
6. *SentiWordNet 3.0 (SWN)* - A word appeared in SentiWordNet (Baccianella et al., 2010) containing scores positive, negative and objective.
7. *SOCAL* - This lexicon is used for calculating semantic orientation (Taboada et al., 2011). For utilizing intensifiers of the lexicon, we used the logic that if both the intensifier and word is positive add their score, if both are negative add their scores and negate, if intensifier is positive and word is negative then subtract intensifier score from word score and finally if intensifier is negative and word is positive then add their score.
8. *NRC Emotion Lexicon* - a list of English words and their association with eight basic emotions and sentiment tags (Mohammad and Turney, 2013). In case of our classifier, we only utilized two polarity tags

For training our supervised classifiers, we used a manually tagged gold-standard dataset containing a total of 1500 training instances, i.e 500 of each polarity, created by merging data from (Mandal and Das, 2018) and (Ghosh et al., 2017b). In case of testing, we used a total of 600 tweets, i.e 200 of each polarity. The data (training and testing) had no data in common in the released versions. However, the features as mentioned for supervised learning were also used to train these classifiers. Different evaluation parameters scored by each of the classifiers are described in Table 3. Other than the *accuracy*, the mean value was considered over the three polarities for each of the other parameters. In Table 3, we can clearly find that SGDC achieved the best F1-Score with a value of 78.70. Thus, for building our polarity tagger, we finally used the trained model of SGDC. Paramaters (*Param*) were Accuracy (*Acc.*), Precision (*Prec.*), F1- Score (*F1*) and G-Score (*G*).

	Naïve Bayes (NB)			**Linear Model (LM)**	
Param	*GNB*	*BNB*	*MNB*	*SGDC*	*LRC*
Acc.	74.83	76.16	78.16	78.66	77.00
Prec.	75.05	76.25	78.56	79.20	77.40
Recall	74.83	76.16	78.16	78.66	77.00
F1	74.87	76.17	78.18	**78.70**	77.02
G	74.90	76.19	78.27	78.81	77.11

Table 3: Performance of different classifiers.

The confusion matrix of the best performing classifier, that is SGDC, is shown in Table 5. We can see that the classifier is quite stable and not very biased towards a single polarity. The best individual polarity accuracy is for neutral tweets (83%), which again supports the point regarding it's stability.

	pos	*neg*	*neu*
pos	161	12	27
neg	17	145	38
neu	13	21	166

Table 4: Confusion matrix of SGDC classifier (italics - predicted values, roman - true values).

The final algorithm we used for sentiment tagging by combining rule based and supervised into a hybrid routine

[10] http://scikit-learn.org/stable/

is described below -

```
Input ← sentence
Output → polarity

Step 1: if FLNG (sentence) ≠ neutral then
        return FLNG (sentence) else goto Step 2
Step 2: if EMO (sentence) ≠ neutral then
        return EMO (sentence) else goto Step 3
Step 3: if HT (sentence) ≠ neutral then
        return HT (sentence) else goto Step 4
Step 4: return SGDC (sentence)
```

Here FLNG, EMO and HT are the functions described under rule based methods in feature section and SGDC is our trained supervised classifier.

4.2. Annotators' Guidelines

As the data is already language and sentiment tagged by the systems, the manual annotation efforts were reduced drastically. However, in order to prepare a gold standard corpus with good quality, we finally handed it over to our annotators along with a number guidelines. We provided a very less number of guidelines as most of the urgent issues were already considered by using our systems.

Language Tagging - In case of language tagging, the scope of the current target word and the words preceding and succeeding the target word were considered.

Bengali (BN) & English (EN) Tag

LG1 The word is present in the respective language dictionary or is a slang or acronym of that language.
e.g. "*hall*" tagged as EN and "*ghor*" tagged as BN.

LG2 whether the word in context belongs to that respective language or not.
e.g. "*bar*" in "*onek bar bolechi*" is tagged as BN.

LG3 The word has any English/Bengali prefix or any English/Bengali suffix.
e.g. "*hall is*" tagged to EN and "*ghor ta*" tagged to BN.

Unknown (UN) Tag

LG4 The word does not belong to Bengali or English.
e.g. "*amr*" is tagged to UN.

LG5 The token is not recognized (like misspelled words).
e.g. "*ankushloveuall*" is tagged to UN.

LG6 The token is a special character, emoticon, URL, etc.
e.g. "@" is tagged as UN.

Sentiment Tagging - For polarity tagging, the authors' perspectives were taken into account and the emotions conveyed from the overall tweet were considered as well.

Positive Tag & Negative Tag

SG1 The tweet clearly expresses the sentiment towards the aspect term, for example a person, group or an object.
e.g. "*Sir, Boss 2 hit movie hobe. Eid ar sera movie.*" is tagged as positive.

SG2 The tweet clearly expresses the polarity state in mind of the author.
e.g. "*Dhurr ar posachhe na all these things.*" is tagged as negative.

SG3 The tweet clearly reports a polar sentiment or mood which may or may not be attributed directly by the author.
e.g. "*@username1 yes ami @username2 dadar pagol fan onek diner.*" is tagged as positive.

Neutral Tag

SG4 The tweet contains a mere observation or mention of an objective fact.
e.g. "*Dure oi yellow building ta holo shopping mall.*" is tagged as neutral.

SG5 It does not particularly convey any state of mind or opinion. A neutral sentiment is expressed towards the aspect term(s).
e.g. "*Cinema ta release koreche.*" is tagged as neutral.

Conflicts - The confusions occurred during annotation were tabulated as follows

English (EN) Tag

1. In the context of a word that contains numerical values were considered by the annotators. For example '11 AM' was tagged as EN by Annotator A while Annotator B tagged "11" as UN and "AM" as EN, separately.
2. Country names were tagged as EN and UN by Annotator A and B, respectively.
3. Universal words such as, "*table*" were tagged as EN by Annotator A and BN by Annotator B.
4. Words such as "*to*" were tagged as both EN and BN depending on the context and their phonetic representations.

Bengali (BN) Tag

1. The role of the suffix in a word was also dealt ambiguously. For example "*film*" is tagged as EN whereas "*film (ta)*" was tagged as BN.

Unknown (UN) Tag

1. Numerical values such as "1", "2" were tagged as UN.

We considered two sets of human annotators A and B along with system as the third set. The inter annotator agreement values or Cohen's Kappa (K) are shown in Table 5 with respect to each pairs of annotators. In case of sentiment tagging, the annotators agreed on majority of the tweets. However, in both language as well as sentiment tagging, the agreement scores between the sets of manual annotators were comparatively better than the agreements that were calculated with respect to systems. One of the reasons that degrades the system results is relatively small set of training instances in case of both language and sentiment tagging. The annotation details of the system and human annotators are shown in Table 6.

Language Tagging - Kappa	
Annotator A-System	0.69
Annotator B-System	0.65
Annotator A-Annotator B	0.83
Sentiment Tagging - Kappa	
Annotator A-System	0.83
Annotator B-System	0.82
Annotator A-Annotator B	0.94

Table 5: Inter annotator agreement.

Training Data			
	Language Tag		
	BN Tag	**EN Tag**	**UN Tag**
System	22801	15130	331
Annotator A	22460	15478	324
Annotator B	22471	15471	320
	Sentiment Tag		
	Pos Tag	**Neg Tag**	**Neu Tag**
System	988	926	586
Annotator A	1010	987	503
Annotator B	1000	1000	500
Testing Data			
	Language Tag		
	BN Tag	**EN Tag**	**UN Tag**
System	22896	12129	421
Annotator A	22418	12620	408
Annotator B	22416	12616	414
	Sentiment Tag		
	Pos Tag	**Neg Tag**	**Neu Tag**
System	1077	642	741
Annotator A	1094	698	668
Annotator B	1090	705	665

Table 6: Annotation details of system and human annotators.

5. Corpus Aspect Analysis

The released data distribution is shown in Table 7. In both training and testing, the quantity of neutral data is comparatively less as we found that most of the tweets we mined had a polarity. Here, we have analyzed different aspects of our developed gold standard data like code-mixing complexity and generic language aspects. Statistics on some of sentiment affecting aspects like polarity word count, emoticons count, etc were also carried out.

Distribution			
Purpose	**Positive**	**Negative**	**Neutral**
Training Data	1000	1000	500
Testing Data	1090	705	665

Table 7: Data distribution.

Language Aspects - Here we analyzed both complexity aspect contributed by code-mixing property (shown in Table 8) as well as other aspects like polarity token counts and mean length (shown in Table 9). Code-Mixing Index (CMI) introduced by (Das and Gambäck, 2014) indicates us the amount of code-mixing found in discourse. Another metric we have calculated which shows the complexity of multilingual corpus is the Complexity Factor (CF) proposed by (Ghosh et al., 2017b). CF takes into account three factors- language (LF), switching (SF) and mix (MF) factors. CF was calculated using all the three methods mentioned in that paper. From Table 8, we have observed that the collected code-mixed data has a higher code-mixing index as compared to FIRE 2015 [11] Shared Task Corpus (CMI = 11.65) and ICON 2015 [12] Shared Task Corpus (CMI = 5.73). Thus, we can conclude that our data is more complex from code-mixing point of view as compared to FIRE and ICON corpus. We can also see that on an average, positive data has higher code-mixing as compared to other polarities while neutral has comparatively lower code-mixing. From the training and testing values we can also see that the variance is quite nominal, thus adding to the quality of prepared corpus.

		Training			Testing		
index	**f**	**pos**	**neg**	**neu**	**pos**	**neg**	**neu**
CMI	min	4.02	4.24	4.20	4.16	4.20	4.18
	max	50.0	48.6	46.2	48.6	48.6	47.5
	mean	31.0	27.9	22.6	23.4	21.6	20.0
CF1	min	0.38	0.52	0.46	0.44	0.46	0.45
	max	20.8	18.0	23.0	37.5	23.0	37.5
	mean	4.20	3.93	3.71	4.14	3.67	4.17
CF2	min	4.58	4.81	4.76	4.63	4.76	4.72
	max	57.5	62.4	54.8	69.2	64.6	69.2
	mean	26.1	24.4	20.5	23.3	21.4	20.7
CF3	min	4.25	4.41	4.36	4.27	4.36	4.31
	max	53.8	58.4	51.8	68.0	61.5	68.0
	mean	24.2	22.6	19.1	21.6	19.9	19.3

Table 8: Complexity statistics (f - function).

Other important language related aspects are are shown in Table 9. The relation for negation count is $\geq$ as lexical checking was done so whereas there might be more number of negations. The aspect values were calculated based on post annotator tagging of language and sentiment. The probable reason for higher negation in negative data is mainly because of the habit of users to express negative sentiment by negating positive words, e.g. *bhalo na* which means "not good". This can be confirmed as well by skimming through the data. The table also tells us that users tend to write relatively more to the point and short tweets while expressing negative sentiments. This is checked from the mean length and UN word count values. Also, BN/EN ratio tells us that users tend to use more Bengali words for expressing objective sentiments.

[11] http://fire.irsi.res.in/fire/2015/home

[12] http://ltrc.iiit.ac.in/icon2015/

Language Aspects					
	Training Data				
N	**Attribute**	**R**	**Pos**	**Neg**	**Neu**
1	Negation Count	≥	148	449	170
2	Mean Length	=	18.50	18.06	17.91
3	BN word count	=	8541	8866	5064
4	EN word count	=	6997	6535	1939
5	UN word count	=	110	93	117
6	BN/EN Ratio	=	1.220	1.356	2.611
	Testing Data				
1	Negation Count	≥	182	375	200
2	Mean Length	=	18.94	16.23	17.46
3	BN word count	=	8664	7388	6364
4	EN word count	=	5985	4329	2302
5	UN word count	=	168	118	128
6	BN/EN Ratio	=	1.447	1.706	2.764

Table 9: Language statistics. (R - relation)

Emotion Aspects - Statistics of sentiment affecting aspects are shown in Table 10. Users tend to explicitly convey their feelings by using the feeling tag more so while expressing negative sentiment as compared to positive. For emoji count the relation is $\geq$ as lexical checking was done, so in reality there might be more number of emoticons. Same is the case for polarity word count, but here $\approx$ is used instead as contextually the word may not be positive or negative. From positive and negative word count in Table 10 we can see that users tend to use English polarity words more often as compared to Bengali while expressing.

Sentiment Aspects					
	Training Data				
N	**Attribute**	**R**	**Pos**	**Neg**	**Neu**
1	POS emoji count	≥	18	2	2
2	NEG emoji count	≥	3	17	1
3	POS word count	≈	1187/ 587	118/ 51	35/ 26
4	NEG word count	≈	103/ 65	757/ 416	32/ 19
5	Feeling tag count	=	5	10	1
	Testing Data				
1	POS emoji count	≥	22	5	3
2	NEG emoji count	≥	6	20	1
3	POS word count	≈	918/ 435	106/ 42	27/ 19
4	NEG word count	≈	119/ 72	673/ 341	28/ 13
5	Feeling tag count	=	4	8	2

Table 10: Sentiment affecting aspects. For POS, NEG word count representation format is EN/BN. (R - relation)

Other Aspects - The most common polarity carrying words from the code-mixed data are shown in Table 11. From the table we can see that the most common polar words are highly polar. These words are commonly used while speaking as well. It can also be seen that a lot of counterparts are present in the table, like bhalo - good, osadharon - special, kharap - bad, betha - pain, etc.

Most Common Words (freq>150)		
	Bengali	**English**
Positive	bhalo, besh, shundor, darun, moja, pochondo, osadharon	love, best, good, comedy, better, special, famous, happy
Negative	kharap, baje, kosto, boka, bekar, chinta, jhogra, betha	poor, bad, problem, old, sad, busy, bogus, pain

Table 11: Some common Bengali and English words, training and testing data combined.

6. System Performance on Final Data

After the final annotation was done we tested our systems again on the new gold-standard data. Both the language tagging system and sentiment tagging system (SGDC) was trained on the training data and evaluated on the testing data. The language tagger performed surprisingly well and got an accuracy of 81%. With the sentiment tagging system we expected a significant improvement due the increased size of the training data. It indeed performed better and got an accuracy of 80.97% and F1-Score of 81.2%. In future we would like to test different feature combinations and add contextual features as well to improve our system.

7. Release Format

The final gold-standard dataset is available in JSON format. We have chosen JSON since it is more compact, lightweight, flexible and easier to use compared to XML. CSV was ignored as well since we needed to represent a hierarchical structure which is much easier with JSON as well. Another problem with CSV is that a standard reader application (e.g. Excel) is quite slow at opening large files as well as unstructured encoded values and spilling. The objects/values provided in the released JSON file are id (data number), lang_tagged_text (language tagged text), sentiment (-1 ← negative, 0 ← neutral, 1 ← positive) and text (without language tag). A single sample from the JSON file is given below -

id: 83
lang_tagged_text: Onekdin\bn por\bn spotlight\en e\bn fire\bn eshe\bn nijeke\bn besh\bn bikheto\bn bikheto\bn lagche\bn ,\un I\en am\en toh\bn very\en hpy\en .\un
sentiment: 1
text: Onekdin por spotlight e fire eshe nijeke besh bikheto bikheto lagche, I am toh very happy.

8. Conclusion & Future Work

In this paper we have described the steps involved in building the system which we have used for collecting and preparing gold-standard Bengali-English code-mixed data for sentiment analysis. To the best of our knowledge, it is the first publicly released data of its kind. The data we present also has a reliable inter-annotator agreement, K - 0.83 for language tag and K - 0.94 for sentiment tag. We also discuss the challenges faced in each step which should be overcome in future for an improved system. In future, we wish to improve the quality of our system by increasing the population size of our resources and training our classifiers on bigger data. We also wish to find a correlation between α (BN token count) and the keyword used for querying to the API so that the value of α can be varied automatically using computationally calculated rules to fetch more relevant data which in this case was Bengali-English code-mixed.

References

Baccianella, S., Esuli, A., and Sebastiani, F. (2010). Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *LREC*, volume 10, pages 2200–2204.

Barman, U., Das, A., Wagner, J., and Foster, J. (2014). Code mixing: A challenge for language identification in the language of social media. In *Proceedings of The First Workshop on Computational Approaches to Code Switching*, pages 13–23.

Das, A. and Bandyopadhyay, S. (2010). Sentiwordnet for bangla. *Knowledge Sharing Event-4: Task*, 2.

Das, A. and Gambäck, B. (2014). Identifying languages at the word level in code-mixed indian social media text.

Ghosh, S., Ghosh, S., and Das, D. (2017a). Complexity metric for code-mixed social media text. *arXiv preprint arXiv:1707.01183*.

Ghosh, S., Ghosh, S., and Das, D. (2017b). Sentiment identification in code-mixed social media text. *arXiv preprint arXiv:1707.01184*.

Joshi, A., Prabhu, A., Shrivastava, M., and Varma, V. (2016). Towards sub-word level compositions for sentiment analysis of hindi-english code mixed text. In *COLING*, pages 2482–2491.

Mandal, S. and Das, D. (2018). Analyzing roles of classifiers and code-mixed factors for sentiment identification. *arXiv preprint arXiv:1801.02581*.

Mohammad, S. M. and Turney, P. D. (2013). Nrc emotion lexicon. Technical report, NRC Technical Report.

Muysken, P. (2000). *Bilingual speech: A typology of code-mixing*, volume 11. Cambridge University Press.

Pak, A. and Paroubek, P. (2010). Twitter as a corpus for sentiment analysis and opinion mining. In *LREc*, volume 10.

Sharma, S., Srinivas, P., and Balabantaray, R. C. (2015). Text normalization of code mix and sentiment analysis. In *Advances in Computing, Communications and Informatics (ICACCI), 2015 International Conference on*, pages 1468–1473. IEEE.

Taboada, M., Brooke, J., Tofiloski, M., Voll, K., and Stede, M. (2011). Lexicon-based methods for sentiment analysis. *Computational linguistics*, 37(2):267–307.

Vyas, Y., Gella, S., Sharma, J., Bali, K., and Choudhury, M. (2014). Pos tagging of english-hindi code-mixed social media content. In *EMNLP*, volume 14, pages 974–979.

Automatic Evaluation of Alignments without using a Gold-Corpus - Example with French-Japanese Aligned Corpora

Raoul Blin
CNRS-CRLAO
105 Bd Raspail, 75006 Paris, France
blin@ehess.fr

Abstract

This paper presents a test designed to automatically evaluate the alignment quality of a bilingual aligned corpus without comparing it to a reference/gold corpus. We justify bypassing a reference corpus and present the advantages of the test, particularly for language pairs where few bilingual aligned corpora exist. The test, which is based on the observation of single-translation words, is demonstrated using three bilingual French-Japanese corpora : a manually translated and aligned corpus, a freely translated and semi-manually aligned corpus, and a freely translated and automatically aligned corpus. This paper shows that the results achieved validate the proposed evaluation technique.

Keywords: Evaluation, Alignment, Japanese, French, OPUS

1. 1 Introduction

There is a great and undoubtedly growing need for bilingual and multilingual aligned corpora. Aligning corpora semi-manually is costly in terms of time and human resources, hence the need for automatic alignments. This poses the question of how we can evaluate the quality of such alignments. Reviewing the literature on this subject (for example (Langlais, Véronis & Simard, 1998)) along with subsequent research (for example Cherry & Lin (2003), Matusov, Zens & Ney (2004), Li, Sun & Xue (2010), etc.) shows that the evaluation method generally adopted involves comparing the aligned corpus to a gold corpus. The gold corpus itself is a semi-manually aligned subset of the corpus being evaluated. Using a gold corpus both impedes and is contradictory to the production of automatically aligned corpora. Indeed, as we have already seen, semi-manually aligning a corpus is costly. It would therefore be more efficient to evaluate alignments without relying on gold corpora.

Existing methods that bypass gold corpora are problematic for several languages. Firstly, manual evaluation is extremely costly if scientific procedures are to be respected (double or triple evaluation by experienced bilingual evaluators followed by calibration and re-evaluation). Secondly, such an evaluation can only be conducted on samples, whereas automatic evaluation is assumed to encompass the entire corpus. The use of identical words (Simard, Foster & Isabelle, 1993 ; Zhang et al., 2005) is only possible for languages employing the same writing system. Lastly, hapaxes (Lardilleux & Lepage, 2008) require large bilingual corpora which unfortunately are lacking in many language pairs, including Japanese-French, our focus here.

In this paper, we suggest a simple method to *evaluate* the quality of alignment by observing single-translation words. Single-translation words (abbreviated henceforth as *stwords*) are words with only one possible translation (Langé & Gaussier, 1995) but are not necessarily monosemic. For example, although “John” denotes many men in the world, it is always translated as “jon” in Japanese (nevertheless, see the discussion in section 2.2). It can therefore be considered an stword. We must insist here that the method is not designed to make alignments, even though it can be used for this purpose. This method can be applied to any kind of bisegment: words, chunks, sentences and so on. In section 2, we set out the principles of the method; section 3 describes the results of an experiment conducted on two French-Japanese aligned corpora; and section 4 consists of a discussion of these results.

We originally used this method to evaluate the very small number of existing Japanese-French bilingual aligned corpora, hence the focus on this language pair here. The linguistic particularity of this pair is that both languages possess different syntactic properties, including word order, and use different writing systems. They also represent a very common situation: namely, that each language has long been well-endowed with monolingual corpora but has few bilingual alignments (compared with Japanese-English or European language pairs, for example). The same applies to most Japanese-European language pairs (except English), where bilingual aligned corpora are rare. In section 4 we suggest ways to apply our procedure to other language pairs.

2. Evaluation Based on the Observation of Single-translation Words

2.1 The Base

The purpose of the tests is to evaluate (not to make) alignments of texts which do not necessarily use same characters, same syntactic structures and which can be short. Also, the texts are neither (necessarily) annotated nor tokenized.

We use two tests, both based on the observation of stwords. Let us consider an aligned bitext. The source text contains stwords; the target text, the translations of these stwords. Ideally, if the corpus has been correctly aligned, any source segment with a given number (N1) of occurrences of an stword should be aligned with a target segment containing the same number (N2) of translations of this stword. The difference between the two numbers is the evaluation criteria. N1>N2 means that at least one sub-segment that contains an occurrence of the translation of the stword is missing in the target segment, and that this sub-segment belongs to another target segment which has been wrongly aligned with another source segment. Conversely, N1<N2 means that a sub-segment (containing a translation of the stword) of another segment is incorrectly aligned with the current source segment. Accordingly, the test consists in counting the number of lines for which N1==N2. The higher the result of this first test, the better the alignment.

Unfortunately, this test is not sufficiently reliable. One reason for this is that many languages substitute special structures (pronouns, etc.) for repeated occurrences of phrases (for example, pronouns are substituted for repeated noun phrases in French). Some languages, like Japanese, elide the repeated phrases. Substitution and elision depend on the distance between occurrences of the phrase. Defining this distance falls outside the scope of this paper. The only thing we can say for sure is that the first occurrence of a proper noun (PN) within a text is systematically translated as word for word. For our purposes, it is these first occurrences of a PN which are reliable. Accordingly, a second test must be conducted. Let us call FO^{src} the first occurrence of a PN in the source text, and NLFO^{src} the number of bisegments in which the source segment contains an FO^{src}. Among these bisegments, let us call NLFO^{tgt} the number in which the target segment contains at least one occurrence of a translation of an FO^{src}. We then obtain (NLFO^{tgt} / NLFO^{src}), which can be considered the recall value.

We need to use stwords that can be extracted without morphosyntactic preprocessing of the bitext. For obvious reasons, we want to avoid manual preprocessing. There are two reasons why we also want to avoid automatic preprocessing. The first is that automatic analyzers can induce wrong POS tagging and consequently errors while evaluating the alignment. The second is that, for most languages, automatic parsing is performed by statistical analyzers that require large training (specialized) monolingual corpora. While this is not a problem for languages like Japanese or French, for others it can be difficult to find efficient analyzers and build suitable corpora.

The reliability of the evaluation depends on the number of stwords in a text, and on the number of segments which contain (first) occurrences of PN. We will provide these values too.

2.2 Proper Nouns as Single-translation Words

Unfortunately, "perfect" stwords do not exist in the case of Japanese-French. However, it is assumed that PNs are very similar to stwords. Let us consider the advantages and disadvantages of PNs with regard the proposed test.

The first advantage is the existence of bilingual Japanese-French lexicons of PNs. Retrieving or building such resources is simple for a wide number of languages.

The second advantage is that for many languages, PNs are morphologically invariable. This implies that finding them in a text should be easy, even without morphological preprocessing. In the case of French and Japanese, there are graphic variations but these are easily predictable, few in number, and can be exhaustively (and automatically) listed. For example, in French, *Tokyo* can be written *Tôkyô* and sometimes *Toukyou*. In Japanese, it is usually written 東京, or perhaps using hiragana (とうきょう) or katakana (トウキョウ). Some PNs can also be written using the Latin alphabet.

The third advantage of PNs is that, provided certain rules are respected, as detailed below, PNs are easy to spot in French and Japanese, and morphosyntactic analysis is not required.

Unfortunately, for our purposes, PNs also have certain drawbacks.

1) Some common nouns are derived from PNs but not their translation. For example, the French CN *Français* ('French people') is translated by a derived PN in Japanese (ex.: *nihon-jin*, 'Japan-people'). The number of occurrences of PNs differs in bisegments that contain such word pairs. For example, consider the following French and Japanese sentences that correspond to the sentence "[French people]$_{CN}$ leave France$_{PN}$". The French sentence contains only one occurrence of the PN/stword *France*, whereas the Japanese sentence contains two occurrences of its translation, *huransu*, because *huransu* also appears in a derived CN.

Les Franç-ais$_{CN}$ *quittent la France*$_{PN}$.
<->
huransu$_{PN}$ *- jin wa huransu*$_{PN}$ *wo deteiku.*
France - peo. TOP France OBJ leave

To avoid such a shift without morphosyntactically preprocessing the text, the simplest solution would be to use the French corpus as the source corpus. This is convenient for other reasons, as set out below. We studied other strategies but rejected them on the grounds that they required morphosyntactic preprocessing of the text or lexicon.

2) Many of the Chinese characters used in Japanese can be both PNs or graphic components of morphs. For example, the character 順 *jun* can occur as a PN (the given name "Jun") or as a graphic (as opposed to semantic) component in the CN 順序 (*junjo*, "order"), among others. However, these two occurrences are unrelated. Such ambiguity most often arises with PNs consisting of one Chinese character. Morphological preprocessing would eliminate such errors but might produce others. Given this, we preferred to simply eliminate all one-Chinese-character PNs from our lexicon.

3) The same ambiguities can arise in French too. For example, *Violette* is ambiguous because it can refer to the flower *violette* or to the girl's name. Once again, in order to avoid errors without resorting to morphological parsing, we simply retrieved the 13,000 French morphemes that can occur as something other than PNs.

4) Use of PNs can differ by language. While French substitutes a pronoun for a repeated PN, Japanese allows for repetition. To account for this phenomenon, we added a second test based on the first occurrence of a PN in a text. In contrast, in Japanese, people are frequently designated by their name, whereas French uses a pronoun

(*tu*, *vous*, etc.). The number of anthroponyms may thus be lower in French than in Japanese, regardless of the quality of the alignment. This is the case in the following dialog with the interlocutor Tanaka.

Ja: *tanaka san wa dô omoimasuka?*
Tanaka Mr TOP how think?
(lit.: 'what Mr Tanaka thinks?')

Fr: *Qu'en pensez-vous?*
What think you?
'What do you think?'

There does not appear to be any way to avoid such errors, or to quantify the shift caused by this phenomenon. This shift will no doubt differ significantly depending on the type of corpus used, with dialogs being particularly affected.

Metonymy also affects results. For example, French frequently designates a governing body by the place where it is located. For example, “Paris refused”, meaning that “France/the French government (located in Paris) refused”. Such use of metonymy is not observed in Japanese, where the above sentence would no doubt be translated as *huransu ga kotowatta*, “France refused”. In this case, if *Paris* occurs in the French segment, its translation will not appear in the corresponding Japanese segment.

4) Some idioms use PNs but cannot be translated word for word. *Doux Jésus!* (meaning “Jesus Christ”) is translated as *masaka!* which is not a PN. Such idioms are well known. To avoid errors, any segments in which they occur can simply be excluded.

3. Experiment

We applied the proposed tests to three aligned bitexts (see quantified descriptions in Table 1). A manual evaluation of the alignments suggests that they differ significantly in quality. If the test is efficient, it should reveal this difference. The first corpus (OPUS-fj) is the Japanese-French sub-corpus of the OPUS project (Tiedemann & Nygaard, 2004; Tiedemann, 2012). This is actually the only freely available, *large-scale* aligned Japanese-French corpus. Several alignment units are used: graphic lines, sentences, syntactic phrases. We assume this corpus to be representative of the attempts to build large-scale aligned corpora automatically. To our knowledge, OPUS-fj has never been evaluated. A manual evaluation suggests that the alignments and translation are of poor quality. For example, in the large bitext OpenSubtitles2013, a one-line offset clearly appears. The second corpus (PUD[1]) is the test-corpus used for CoNLL 2017 shared task on Multilingual Parsing from Raw Text to Universal Dependencies. It is manualy aligned and translated by professionals. The alignment unit is the sentence. Unfortunately, it is small (only 1,000 sentences). We then use a third corpus: ALIGNJaFr_BABT-0.2_specialEval. It is composed of (semi-)professional translations that have been aligned automatically and manually corrected. Depending on the bitext of the corpus, the alignment unit may be the sentence or syntactical phrases. Assuming the test is efficient, the best score should be obtained with PUD, followed by ALIGN. The worse score should be obtained with OPUS-fj.

We used a lexicon (NP-fj.v0.2) made of 172K pairs of Japanese and French PNs, extracted from JaLexGram-v0.25 . 858 PNs written with only one character were excluded. We used Lefff (Sagot 2010). to exclude 731 Japanese PNs that can be mistaken for other part of speech.

In both tests, French is the source language and Japanese the target language. To count the PNs in a bisegment, we count all the PNs from the French segment, look for their translations in the list of PNs, and then count the occurrences of those translations in the target segment. In order to extract the PNs from the French text, we locate any words that begin with a capital letter and retain those that appear on the list of PNs.. When extracting a translation in Japanese, we simply look for a substring equal to this translation.

3.1 Test 1 : Observation of all Occurrences of Stwords

To evaluate the quality of alignment, we focus on the bisegments in which the source segment contains at least one PN. Let us call nbbiseg the number of such bisegments in a text. For each bisegment, let's call $PNsrc_i$ a PN which occurs in the source segment, $PNtgt_i$ its translation, occ(W) the number of occurrences of a word W in the segment (not the bisegment) where it occurs. For example $occ(PNsrc_i)$ is the number of occurrences of $PNsrc_i$ in the source segment. A “good” $PNsrc_i$ is such that $occ(PNsrc_i)= occ(PNtgt_i) =/= 0$.

We then calculate the proportion of “good” (source) segments which include only good PNsrc. The result can be interpreted as the recall score:

$$\frac{\text{number of “good” segments}}{\text{nbbiseg}}$$

	OPUS-fj	ALIGN	PUD
Nb of words (French)	12,672,676	202,687	20,543
Nb of bisegments	1,868,319	10,821	1,000
% of src segments with PN(s)	1.94	19.72	19,08
% good segments	56.40	72.54	91.62

Table 1: Results of test 1

Because ALIGN and PUD have been manually aligned, they should provide similar scores. The scores for ALIGN are lower for many reasons relating to translation rather than alignment. ALIGN-French frequently uses metonymy. For example, the 87 occurrences of “Washington” all refer in fact to the United States, not to the city. In ALIGN-Japanese, this word has therefore been translated as *amerika* or *beikoku* , not as *washinton*.

[1] Files for Japanese and French are downloadable from https://github.com/UniversalDependencies/

3.2 Test 2 : Observation of the First Occurrence of each Stword

The second test is similar to the previous one but only takes into account the first occurrence of a PN in each text (see the explanation of the counting method in section 2.1). nbbisegO is the number of bisegments in which the source segment contains only first occurrences of PNs. There is a slight difference in the definition of “good PNs”: in test 2 a good PNsrc_i is such that $\text{occ}(\text{PNsrc}_i) \geq \text{occ}(\text{PNtgt}_i) > 0$. We thus accept that a PN is translated only once in the target segment (the other occurrences can be elidated or replaced by pronouns). We then provide the percentage of “good” segments:

$$\frac{\text{number of “good” segments}}{\text{nbbisegO}}$$

	OPUS-fj	ALIGN	PUD
% of segments with at least one first occ.	0.15	4.97	11.49
% good segments	50.31	82.71	89.57

Table 2: Results of test2

3.3 Synthesis

We combine the above results in two scores. For each corpus, the first score is the average of test 1 and test 2. To emphasize the results obtained with the first occurrences of stwords when manipulating languages which do not repeat PNs, we provide a second score : (SPN+(2*SPNO))/3.

	OPUS-fj	ALIGN	PUD
% good segments (average)	53.35	77.62	90.59
Average (emphasize test 2)	52.34	79.32	90.25

Table 3: Synthesis of test 1 and 2.

For both tests using OPUS-fj, the values significantly differ depending on the sub-corpus. However, the overall score of OPUS-fj was dragged down by the low score of large subcorpora like Open Subtitle. Perhaps some sub-bitexts in OPUS-fj have been automatically (incorrectly) translated (see the discussion of this problem in automatically building corpora in Ruopp & van der Meer (2015)).

3.4 Reliability of Extraction Method

One particularity of our method is that it does not involve preprocessing the corpora. We carried out a qualitative comparison of this method with manual extraction and automatically POS-tagged texts. For this purpose, we compared extraction procedures using our method with a French corpus POS-tagged with TreeTagger (Schmid,1995) and a Japanese corpus POS-tagged with Mecab (Kudo, 2006) with the dictionnary mecab-jumandic[2]. Both TreeTagger and Mecab are commonly used in NLP. We used a test-corpus consisting of 100 sentences randomly extracted from ALIGN (version 1).

French: there were no errors using our extraction method of *known PNs*. But compound and unknown PNs are not take into account. TreeTagger take into account unknow PNs but not compound PNs. It encountered several errors. For example, most of the non-PN words positioned at the beginning of a sentence with a capital letter were wrongly interpreted as PNs, including adverbials like *malgré* “despite”. Thus, preprocessing with Treetagger do not necessarily improve extraction from French.

Japanese: there were no errors using our extraction method but many occurrences of PNs were overlooked. By excluding PNs with one character, we missed 16 occurrences of translations. On the other hand, Mecab correctly analyzed all these short PNs. In addition, as we predicted, with both methods country names (ex.: *huransu* “France”) were retrieved from the derived CN (*furansu-jin* “French people”). Mecab made errors on 7 nominal morphs which have been analyzed as PNs. It does not take into account compound PNs.

4. Discussion

As we can see in Table 3, for both tests best values were obtained with the manually aligned corpus PUD, followed by ALIGN, and then by the automatically-aligned corpus OPUS-fj. These results are in line with our expectations. We therefore assume that despite their simplicity, the two tests provide a reliable measure of alignment quality, even without using gold corpus.

We provided three scores that can be used differently. Test 2 produces the most reliable score because it is less sensitive to the syntactic and pragmatic differences between French and Japanese. However, it uses only part of the stwords. Its efficiency is therefore low when it is applied to corpora containing few stwords. For corpora of this kind, we prefer test 1, despite it being less reliable. Some people may prefer to have a single score rather than manipulate two scores. In such situations, we suggest a simple synthesis obtained by calculating the average of both tests. Of the two resulting scores obtained with this method, we suggest using the score that emphasizes the more reliable test 2.

The evaluation of alignments by hand or using gold corpora is reliable enough to be self-sufficient. However, the test proposed here no doubt has some weak points. Reliability depends on the frequency of the PNs and on the exhaustivity of the lexicon. While the test provides good information, it may not be reliable enough. It should be used in conjunction with other tests.

We are currently exploring additional ways of evaluating alignments without using a gold standard corpus or resorting to manual evaluation. One method is based on words written with Latin characters. These are easy to extract from Japanese and make it possible to evaluate the alignment with Japanese as the source language. The other evaluation method we are currently exploring is based on automatic translation created by systems trained on the corpus to be evaluated. Unfortunately, it is only applicable to large corpora. Ultimately, we expect to provide a global score based on the three tests.

[2] mecab-jumandic 5.1.20070304-7

Although we evaluated the test using French-Japanese aligned corpora, it can be applied to many other language pairs where the target language is Japanese. To avoid pre-processing the source language, the target language has to conform to at least two requirements. First, words need to be graphically separated. Second, PNs must be graphically distinguished from other words. For example, in French, PNs are marked by a capital letter. Such a mark is inefficient in German, where CNs also begin with a capital. Of course, a bilingual lexicon would also be necessary.

5. Bibliographical References

Cherry, C. and Lin, D. (2003). A Probability Model to Improve Word Alignment. In Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, Sapporo (JA), pp.88-95.

Kudo, T. (2006). MeCab: yet another part-of-speech and morphological analyzer. http://mecab.sourceforge.net

Langlais, P. Véronis, J. and Simard, M. (1998). Methods and practical issues in evaluating alignment techniques ». In Proceeding of 17th international conference on Computational linguistics, Montréal (CA), pp. 711--717.

Lardilleux, A. and Lepage, Y. (2008). A truly multilingual, high coverage, accurate, yet simple, sub-sentential alignment method », in The 8th conference of the Association for Machine Translation in the Americas (AMTA 2008), Waikiki, Honolulu, United States, pp. 125-132.

Langé, J.-M. and Gaussier, É. (1995). Alignement de corpus multilingues au niveau des phrases », *TAL*, 36(1-2).

Li, P. and Sun, M. and Xue, P. (2010). Fast-Champollion: A Fast and Robust Sentence Alignment Algorithm. In Coling 2010, Beijin, pp.710-718.

Matusov, E. and Zens, R. and Ney, H. (2004). Symmetric Word Alignments for Statistical Machine Translation. In Proceedings of Coling 2004, Geneva (CH), pp. 219--225.

Ruopp, A. and von der Meer, J. (2015). TAUS Moses MT Marquet Report, TAUS, 2015.

Sagot, B. (2010). The Lefff, a freely available and large-coverage morphological and syntactic lexicon for French. In Proceedings of the 7th international conference on Language Resources and Evaluation, Istanbul, Turkey.

Schmid, H. (1995). Improvements In Part-of-Speech Tagging With an Application To German. In Proceedings of the ACL SIGDAT-Workshop, pp. 47–50.

Simard, M. Foster, G. F. and Isabelle, P. (1993). Using Cognates to Align Sentences in Bilingual Corpora. In Proceedings of the 1993 Conference of the Centre for Advanced Studies on Collaborative Research: Distributed Computing - Volume 2, Toronto, Ontario, Canada, pp. 1071–1082.

Tiedemann, J. and Nygaard, L. (2004). The OPUS corpus - parallel & free. In Proceedings of the Fourth International Conference on Language Resources and Evaluation, Lisbon, Portugal.

Tiedemann, J. (2012). Parallel Data, Tools and Interfaces in OPUS. In Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12), Istanbul, Turkey, 23-25.

Zhang, Y. and Liu, Q. and Ma, Q. and Isahara, H. (2005). A Multi-aligner for Japanese-Chinese Parallel Corpora. In The Tenth Machine Translation Summit Proceedings, 133-140.

6. Language Resource References

ALIGNJaFr_BABT-0.2_specialEval, https://sharedocs.huma-num.fr/wl/?id=kPbOl7luxaq9GtVPGqhg9jRDB3DGKKoq

JaLexgram-v0.25 (2017). https://sharedocs.huma-num.fr/wl/?id=5Y12iTTm0zyVCQ7cmRUcb8mo2FP4U VsZ

NP-fj.v0.2.csv (2017). https://sharedocs.huma-num.fr/wl/?id=t7iRrYjqzIdYrIs50yqc5eaNDmJbAtQ8

OPUS-jafr-20170906.1 (2017).https://sharedocs.huma-num.fr/wl/?id=XmI18ZR6CXBC9XZSHiJEHB4VQ5IrKS8V

Acknowledgements

Thanks to Yves Lepage, Waseda University (tests on OPUS-fj with Moses and Giza++).

Parallel Speak-Sing Corpus of English and Chinese Songs for Speech-to-Singing Voice Conversion

Karthika Vijayan, Haizhou Li
Department of Electrical and Computer Engineering
National University of Singapore, Singapore
{vijayan.karthika, haizhou.li}@nus.edu.sg

Abstract

We present a continuing data collection effort towards building a rich database of English and Chinese pop songs for efficient speech-to-singing (STS) voice conversion. Parallel recordings of lyrics of songs, sung and read by professional singers, are recorded in a professional studio environment using high quality recording equipments under the supervision of a trained sound engineer. Sentence-level and word-level labeling of the read and sung audio files are performed manually. Then temporal alignment between frames of words in the read lyrics and singing is performed automatically using dynamic time warping (DTW) with carefully crafted features. The accuracy of temporal alignment of frames of speech and singing voices is crucial for STS conversion, as it decides the effectiveness of mapping of parameters from speech signals to those of singing. The temporally aligned frames of speech and singing voices are used to map characteristics for STS conversion. The presented database of parallel recordings of speaking and singing voices of same linguistic content assist in facilitating efficient STS conversion, in addition to providing valuable resorts to singing voice analysis, understanding differences in production-perception of speech and singing voices and, evaluation of singing quality.

Keywords: Singing voice analysis, Speech-to-singing, Parallel speak-sing corpus

1. Introduction

The speech-to-singing (STS) voice conversion is a relatively recent application, gaining momentum now-a-days due to the extensive interest from the entertainment industry. In STS conversion, the read lyrics of a song is converted to perfect singing, while retaining the speaker identity of the person reading the lyrics. This task involves mapping the prosody of read lyrics to that of singing, preserving the timbre of the read speech (Saitou et al., 2007a),(New et al., 2010). The STS conversion finds numerous applications related to training and evaluation of singing skills of music students or amateur singers, beautifying singing in karaoke systems, music compositions, singing voice analysis and modeling, better understanding of the relationship between speaking and singing voice styles, etc (Vijayan et al., 2017). Hence building resources to devise efficient schemes for STS conversion will benefit various aspects of singing voice processing.

The basic technique behind STS conversion is demonstrated in the Figure 1. The perfect singing is synthesized by combining the melody of singing extracted from the prosody of a reference singing template/musical score and, speaker characteristics of the person reading the lyrics extracted from the timbre of read speech (Saitou et al., 2007b), (New et al., 2010). The reference singing template consists of a professional singer rendering high quality singing vocals and, the reference musical score consists of the target melody of singing denoted in standard MIDI format. Either reference singing template or target musical score is required for deriving the melody of singing for STS conversion. In this paper, we consider that the reference singing template from a professional singer is available for template-based STS conversion. The linguistic content in the read lyrics and reference singing template are temporally aligned to ensure that correct timbre of a frame of read lyrics is being combined with the prosody from corresponding frames of reference singing template. Later the prosody and timbre from aligned frames of reference singing template and read lyrics, respectively, are combined together to produce synthesized singing in an STS conversion system.

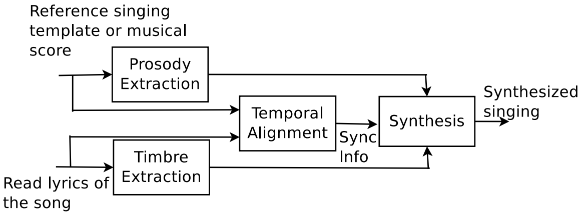


Figure 1: Basic technique for STS conversion.

The accuracy of the synchronization information (Sync Info) from temporal alignment plays crucial role in deciding the accuracy of synthesis. The problem of temporal alignment of linguistic content of read lyrics and reference singing template is not a straight-forward problem. Even though the linguistic content in the read lyrics and reference singing template is the same, the signals corresponding to them vastly differ from each other. The spoken and sung vocals produced by human voice production mechanism exhibit many similar characteristics due to the similarity in the vocal tract system producing them. But there exist some unique properties of the vocal tract system configuration and excitation source, that distinguish between the speaking and singing voice styles (Lindblom and Sundberg, 2007). Due to these differences between speaking and singing, the problem of temporal alignment between read lyrics and reference singing template becomes very challenging.

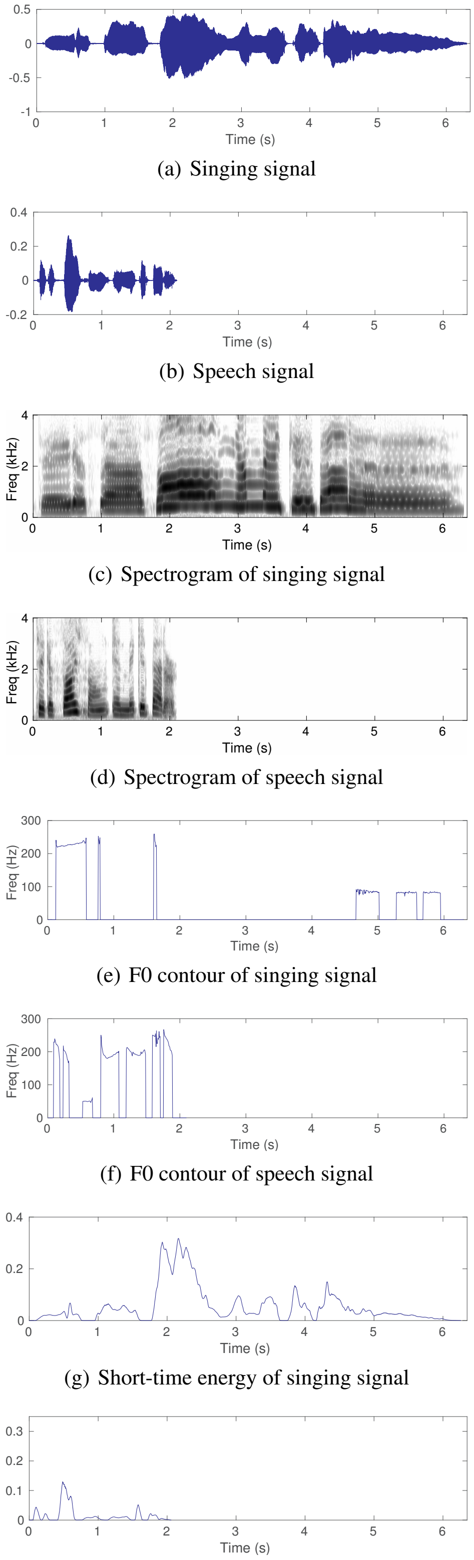
In this paper, we present a parallel speak-sing database, which can provide rich resource to learn the prosody characteristics from reference singing templates and timbre characteristics from read lyrics for efficient STS conversion. We explain an ongoing data collection process in which professional singers are hired to sing good quality song vocals and also read the lyrics of the songs in natural manner. We then proceed to explain the temporal alignment of frames of read speech to the corresponding frames in reference singing template. The resultant database of English songs will either be released in public domain or be shared upon request, tentatively by the end of March, 2018.

The rest of the paper is organized as follows: In Section 2., we explain the differences between speech and singing voice styles that make the process of temporal alignment between them a difficult task. We also discuss the significance of the parallel speak-sing database in STS conversion and details the requirement for accurate temporal alignment between read lyrics and reference singing template. Section 3. explains the parallel speak-sing database under preparation. In Section 4., we elaborate the method devised for effective temporal alignment of read lyrics and reference singing template. In Section 5., we summarize the contributions of this paper towards STS conversion and indicate other significant applications of the presented database, in addition to STS conversion.

2. Speaking and Singing Voice Styles

Vocal sounds are produced as the response of a time-varying vocal tract system to a time-varying glottal excitation signal (Rabiner and Schafer, 1978). Spoken and sung vocals are produced by the same voice production mechanism. Hence they exhibit several similar characteristics like, the consistency in lower order formants. However, the voice production characteristics vary considerably across speaking and singing. A prominent difference between speaking and singing is in the duration of phones. It can be observed that the voiced sounds are largely elongated or compressed depending upon the melody of singing, whereas, the unvoiced sound durations are relatively retained in singing with respect to those in speaking. Also, the dynamic range of singing amplitude is much larger than the same in speaking, resulting in the vast difference in energy of singing voice with respect to spoken sounds (Titze and Sundberg, 1992).

The spectral characteristics of singing vocals differ from those in speaking. Particular positioning of larynx while producing loud singing results in clustering of higher order formants to form the 'singing formant', which serves as the major resource for production of loud and high pitch singing (as in Opera) without the need for increasing the subglottal pressure beyond the capacity of a human singer. The characteristics of excitation source also change vastly in singing. The fundamental frequency of glottal vibrations (F0), equivalently the pitch, is aided by the target melody of singing. But, the pitch in speaking stays relatively flat. As the pitch and/or loudness of singing increases, the subglottal pressure increases as well (Sundberg et al., 1993), (Sundberg et al., 2005), (Sundberg, 2009). To summarize, the major differences between speaking and singing voice styles are,



(a) Singing signal

(b) Speech signal

(c) Spectrogram of singing signal

(d) Spectrogram of speech signal

(e) F0 contour of singing signal

(f) F0 contour of speech signal

(g) Short-time energy of singing signal

(h) Short-time energy of speech signal

Figure 2: Difference between speaking and singing voices.

- Duration
- Energy

- Pitch
- High frequency spectrum (singing formant)

The differences between singing and speaking in terms of these factors can be observed in Figure 2.

2.1. Significance of a parallel speak-sing database

The differences between speech and singing signals as discussed above, present several difficulties in STS conversion such as mapping of dissimilar characteristics, automatic temporal alignment, etc. A parallel speak-sing database of same linguistic content from the same singer can be proven beneficial in such scenario. As the same professional singer sings and reads the lyrics of a song, the speaker identity is preserved in the parallel recordings. This can provide additional advantages, alongside to the same linguistic content, for accurate automatic temporal alignment. Once the read lyrics and reference singing template from the professional singer are aligned, the temporal alignment of read lyrics by an unknown user to reference singing template will be reduced to a speech-speech alignment problem (Vijayan et al., 2017). This problem can be dealt with effectively using dynamic time warping (DTW). Also, the parallel recordings of read lyrics and reference singing template will render an easy resource for learning the speaker-dependent mapping of characteristics from speech signals to singing signals. Thus the parallel speak-sing database construction effort is valuable for designing an efficient STS conversion system.

3. Parallel Speak-Sing Database

In this section, we detail an ongoing data collection effort for designing an efficient STS conversion system. We record singing and speech audio signals in a professional studio environment, employing high quality recording devices under the supervision of a trained and experienced sound engineer. We hired professional singers who either have a diploma in vocal training or have an experience of more than three years in public singing. We provide them with a list of English pop songs and Chinese pop songs, from which they choose 10 songs according to their singing capabilities and vocal range. Special attention was taken to choose singers who can speak and sing in English language without prominent mother tongue influence. The list of English songs provided to the singers is given in Table 1.

The recording of read lyrics is performed by instructing the singers to read the lyrics of songs in their natural speaking manner, without taking long pauses in between. The recording of singing vocals is performed in synchronization with the background music of the corresponding song. The background music is played via headphones to the singer and, each singer is instructed to sing vocals with respect to the background score. Multiple takes are recorded whenever necessary to ensure the pronunciation is correct, the vocals are in-tune with the background music, etc. Currently, two male and two female singers have completed their recording of read lyrics and singing vocals corresponding to three English songs, namely, ‘I dont want to lose you’, ‘Stars shining bright above you’, and ’Fly me to the moon’.

The continuing effort of data collection aims at recording the read lyrics and singing vocals by five male and five female singers, each recording 10 songs. Thus we aim to build a rich database of atleast 100 songs each, from English and Chinese pop genre. This database will aid our ongoing research of implementing an efficient STS conversion system. A similar database of parallel recordings of spoken lyrics and singing vocals can be found here (Duan et al., 2013).

4. Temporal Alignment

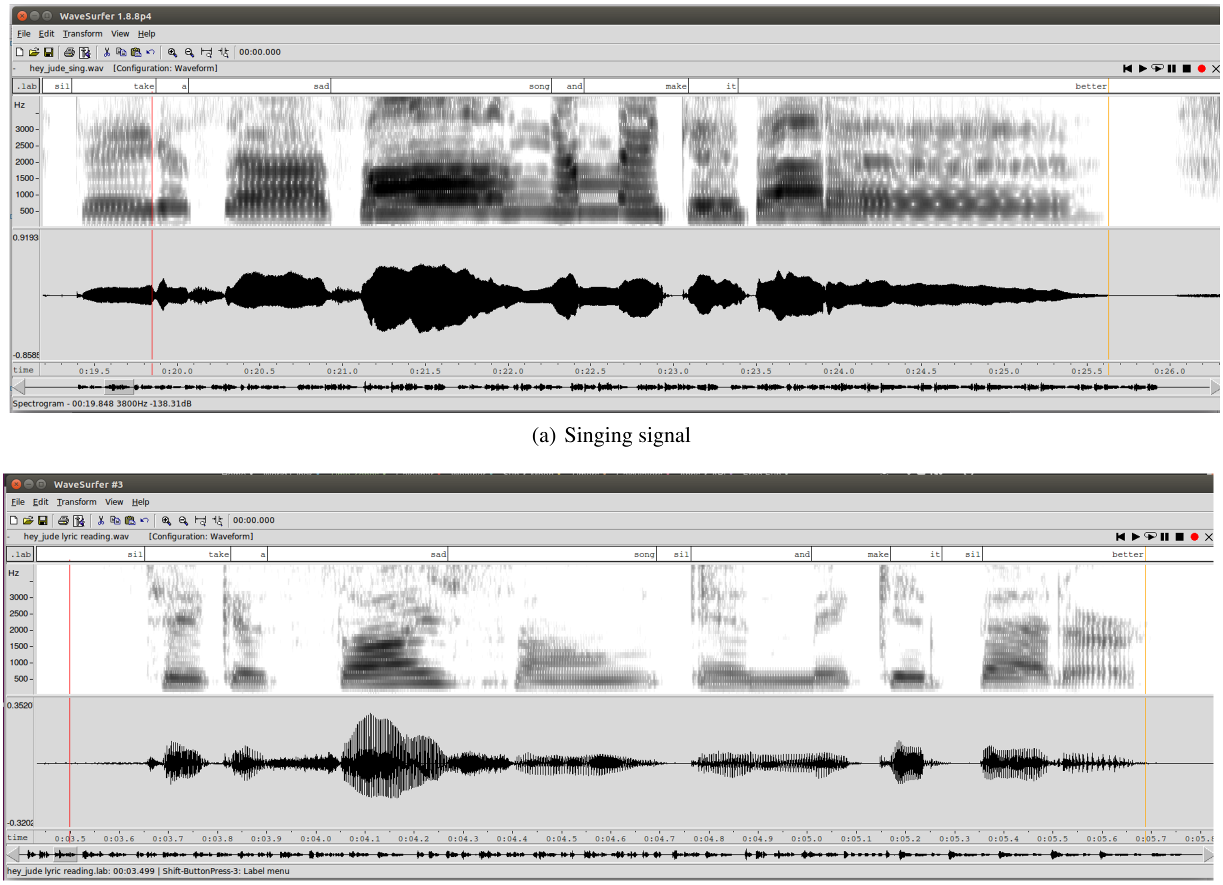
As a part of our database development, we attempt to provide the temporal alignment between read lyrics and singing vocals. The recorded singing vocals will become the reference singing templates for STS conversion. The temporal alignment between read lyrics from an unknown speaker/user of the STS conversion system (user speech) and the reference singing template is extremely crucial for the effectiveness of the STS conversion. The accuracy of temporal alignment will have decisive role in the mapping of characteristics from speech signals to singing. Any error in temporal alignment of linguistic content between speech and singing will result in mismatched combination of timbre from user speech and prosody from reference singing template, consequently producing annoying distortions in the synthesized singing. In our data collection effort, we attempt to temporally align the ready lyrics by the professional singer to the reference singing template. The read lyrics by the singer can act as a bridge between the user speech and the reference singing template in run-time STS conversion (Vijayan et al., 2017).

We perform manual labeling of the read lyrics and singing vocals, at sentence-level and word-level. Researchers and students from our lab, who have working knowledge of speech processing, inspected the waveforms and spectrograms of audio recordings using the tool wavesurfer (Sjolander and Beskow, 2000). They manually label the sentence- and word-level boundaries, creating the required label files. The screenshots of word boundary labeling using wavesurfer for read lyrics and singing vocal are shown in Figure 3.

Once the word-level boundaries are accurately marked, we attempt the frame-level alignment of words from read lyrics to singing vocals. The DTW algorithm was employed for temporal alignment between the speech and singing signals (Sakoe and Chiba, 1978). Generally mel frequency cepstral coefficients (MFCC), together with the delta and acceleration values are used as features of speech signals for DTW alignment. We note that the delta and acceleration values represent the dynamic characteristics of speech signals and, these values are varying vastly across speaking and singing voice styles. Hence we choose to neglect the delta and acceleration values from our feature set, as they will adversely affect the accuracy of temporal alignment. Instead of using MFCC features, we perform a 24th order cepstral analysis from 1024-point Fast Fourier Transform (FFT) corresponding to the read lyrics and singing vocals. Then we choose to retain the first 12 coefficients, termed as low-time cepstral

Table 1: The list of English pop songs included in the database.

S. No:	Song	Artist	Year
1	I Will Always Love You	Dolly Porton	1973
2	My Heart Will Go On	Celine Dion	1997
3	Linger	The Cranberries	1993
4	How Do I Live	LeAnn Rimes	1997
5	Foolish Games	Jewel	1995
6	Billie Jean	Michael Jackson	1983
7	Total Eclipse of the Heart	Bonnie Tyler	1983
8	Take My Breath Away	Berlin	1986
9	Poker Face	Lady Gaga	2008
10	Let it be	The Beatles	1970
11	I don't want to lose you	Tina Turner	1989
12	Staying Alive	Bee Gees	1977
13	Dancing Queen	ABBA	1976
14	How Deep is Your Love	Bee Gees	1978
15	You Light Up My Life	Debby Boone	1977
16	Hey Jude	The Beatles	1968
17	Stars shining bright above you	Ozzie Nelson	1931
18	Fly me to the moon	Kaye Ballard	1954
19	Yesterday	the Beatles	1965
20	Stand by Me	Ernest Tubb	1965



(a) Singing signal

(b) Speech signal

Figure 3: Word-boundary labeling using wavesurfer.

coefficients (LTCC), as the features for DTW alignment. Notice that the low-time cepstrum denotes the vocal tract system properties and high-time cepstrum denote the excitation source characteristics of the voice production system. As the source characteristics are expected to vary vividly across speech and singing voice styles, we choose to isolate out the high-time cepstrum, thereby retaining only the contribution of vocal tract system in the feature set. This strategy helps to preserve the consistent properties of vocal tract system across speech and singing, and rule out the inconsistent properties of excitation source.

The resultant temporal alignment of words between read lyrics and singing vocals, using DTW algorithm with LTCC features, was evaluated on the database presented in (Duan et al., 2013). The word-boundary alignment errors of the automatic alignment were computed against the manually marked transcriptions available with this database. It was observed that the proposed temporal alignment scheme was providing near-accurate synchronization information required for efficient STS conversion.

5. Conclusions

In this paper, we presented an ongoing data collection effort to record parallel speak-sing corpus for English and Chinese pop songs. This database is expected to aid the development of an efficient STS conversion system. The audio recordings of read lyrics and singing vocals are performed in a professional studio environment and trained professional singers are hired to sing the songs and read the lyrics in natural manner. The sentence-level and word-level transcriptions of the audio recordings are labeled manually. Later, automatic temporal alignment between frames within words of read lyrics and singing vocals is performed using DTW algorithm with low-time cepstral features. As temporal alignment is a crucial requirement for a successful STS conversion system, the accuracy of frame-level alignment of words in read lyrics to singing vocals is enhanced using DTW with specially designed features.

Apart from STS conversion, the parallel speak-sing corpus can be proven as a valuable resource for singing voice analysis and understanding. The parallel recordings of same linguistic content by the same speaker in reading and singing voice styles can provide rich resorts to understand the differences in production and perception of speech and singing signals. Thus, the presented database will be advantageous in development of new modeling strategies for singing voices. Also, it can assist in singing quality evaluation, vocal training of music students, etc.

6. Bibliographical References

Duan, Z., Fang, H., Li, B., Sim, K. C., and Wang, Y. (2013). The nus sung and spoken lyrics corpus: A quantitative comparison of singing and speech. In *2013 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, pages 1–9, Oct.

Lindblom, B. and Sundberg, J. (2007). The human voice in speech and singing. In *Springer Handbook of Acoustics*, pages 703–746. Jan.

New, T. L., Dong, M., Chan, P., Wang, X., Ma, B., and Li, H. (2010). Voice conversion: From spoken vowels to singing vowels. In *2010 IEEE International Conference on Multimedia and Expo*, pages 1421–1426, July.

Rabiner, L. R. and Schafer, R. W. (1978). *Digital Processing of Speech Signals*. Prentice-Hall, Englewood Cliffs, NJ, USA.

Saitou, T., Goto, M., Unoki, M., and Akagi, M. (2007a). Speech-to-singing synthesis: Converting speaking voices to singing voices by controlling acoustic features unique to singing voices. In *2007 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 215–218, Oct.

Saitou, T., Goto, M., Unoki, M., and Akagi, M. (2007b). Vocal conversion from speaking voice to singing voice using straight. In *INTERSPEECH*, pages 4005–4006.

Sakoe, H. and Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49, Feb.

Sjolander, K. and Beskow, J. (2000). Wavesurfer - an open source speech tool.

Sundberg, J., Titze, I. R., and Scherer, R. (1993). Phonatory control in male singing: A study of the effects of subglottal pressure, fundamental frequency, and mode of phonation on the voice source. *Journal of Voice*, 7(1):15 – 29.

Sundberg, J., Fahlstedt, E., and Morell, A. (2005). Effects on the glottal voice source of vocal loudness variation in untrained female and male voices. *The Journal of the Acoustical Society of America*, 117(2):879–885.

Sundberg, J. (2009). Voice source studies of register differences in untrained female singing. *Logopedics Phoniatrics Vocology*, 24:76–83, July.

Titze, I. R. and Sundberg, J. (1992). Vocal intensity in speakers and singers. *The Journal of the Acoustical Society of America*, 91(5):2936–2946.

Vijayan, K., Dong, M., and Li, H. (2017). A dual alignment scheme for improved speech-to-singing voice conversion. In *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA-ASC)*, December.

Automatic Acquisition of Opinion Words from Myanmar Facebook Movie Comments

Win Win Thant, Kiyoaki Shirai
University of Information Technology (UIT), Japan Advanced Institute of Science and Technology (JAIST)
winwinthant@gmail.com, kshirai@jaist.ac.jp

Abstract

This paper presents a method for the automatic acquisition of new Myanmar opinion words based on syllable n-gram frequencies, since the Myanmar language uses a syllabic writing system. There is a need for the automatic construction of opinion lexicons for any specific domain, and the proposed method makes it possible to construct an opinion lexicon for Myanmar movies using a bootstrapping approach. We use Myanmar movie comments from Facebook, a popular social network, as a corpus, and determine the valence or polarity of the comments (positive or negative) using a small number of opinion words. These opinion words are then extracted from polarity-identified Facebook comments. Our proposed method is based on n-grams of syllables without word segmentation, since Myanmar is an under-resourced language and no public tool exists for word segmentation.

Keywords: Opinion lexicon, Myanmar movie comments, n-gram syllables, bootstrapping

1. Introduction

Textual information can be divided into two main domains: facts and opinions. While facts focus on the transmission of objective data, opinions express the sentiments of their authors. Opinions are also subjective expressions that describe people’s emotions, appraisals or feelings toward entities, events and their properties. The quantity of users’ comments is increasing daily, since most people enjoy giving their opinions on the Web. The concept of an opinion is very broad, and yet is so important that whenever we need to make a decision we ask for others’ opinions. This is not only true for individuals but also for organizations.

All of us tend to infer the meaning of opinion leaders in our way. For some, opinion leaders are great people such as Nelson Mandela, Mother Teresa and Mahatma Gandhi, who paved the way for revolutions and completely changed the faces of entire countries with their opinions and successful actions. In today’s changing world, social media platforms have taken on the role of opinion leaders. Advertisers use social media-based and celebrity opinion leaders to carry and “trickle down” their message through social media such as Facebook or Twitter, to influence their target groups.[1]

In Myanmar, most people use social media, and especially Facebook, to express their opinions about specific topics in the Myanmar language. Moreover, most of the popular Myanmar film and movie production companies have set up social media accounts to allow users to express their personal opinions about movies. Customer comments are valuable, and are an important source of data for multiple purposes. The reason for using Facebook movie comments in this study is that they provide good material for analyzing the subjectivity and opinions of users. Due to a lack of domain-dependent opinion lexicons, we propose a method for the automatic extraction of opinion words within the domain of movies. To our knowledge, no prior work has been done exclusively on the automatic acquisition of new opinion words using Facebook movie comments in Myanmar.

The remainder of this paper is organized as follows. Section 2 presents related work, focusing on methods of lexicon creation, n-gram statistics and Myanmar syllable segmentation. Section 3 explains the proposed system in detail. The experimental results are described and analyzed in Section 4. We conclude with a summary of the proposed technique and future work in Section 5.

2. Related Work

Our work is inspired by a word-level classification model for automatically generating a Twitter-specific opinion lexicon from a corpus of unlabeled tweets (Bravo-Marquez et al., 2015a). These authors proposed a distributional representation for words by treating them as the centroids of the tweet vectors in which they appear. Bravo-Marquez et al. (2015b) extended a supervised method for expanding an opinion lexicon in the context of emoticon-annotated tweets, by creating a lexicon with disambiguated POS entries and a probability distribution for positive, negative, and neutral classes. In a subsequent paper, these authors proposed a methodology for expanding the NRC word-emotion association lexicon based on a collection of unlabeled tweets (Bravo-Marquez et al., 2016). They compared different word-level features extracted from unlabeled tweets such as unigrams, Brown clusters, POS tags, and word2vec embedding. In contrast to their previous work, this methodology enabled the identification of emotional words from any domain-specific collection using unlabeled tweets.

Our approach is similar to a method proposed by Nagao et al. (1994), who developed a new method and software for n-gram frequency calculation for values of n up to 255; they also automatically calculated n-grams for several large texts in Japanese, including between two and thirty million characters and derived words, compound words and collocations. Here, we use a collection of Facebook comments as a corpus and Myanmar syllables as input for n-grams. Many studies of lexicon creation have been carried out in other languages such as Portuguese and German. Souza et al. (2011) proposed the integration of different linguistic resources to identify opinion-bearing terms, and to create a single opinion lexicon for the Portuguese language. Remus et al. (2010) described the

[1] https://www.facebook.com/mccollinsmedia/

structure of a publicly available German-language resource for sentiment analysis called SentiWS, three sources including the General Inquirer (GI) lexicon, a co-occurrence analysis of rated product reviews and the German collocation dictionary that was utilized to assemble this, and a semi-supervised method used to weight the strength of the entries.

A final line of related work concerns syllable segmentation. In addition to the methods described in Section 1, which use syllabic input, some methods of syllable segmentation have been read for syllabic input. Zin and Mikami (2008) proposed a rule-based approach to a syllable segmentation algorithm for Myanmar text. They created the segmentation rules based on the characteristics of Myanmar syllable structure, but did not consider the non-Myanmar characters within the script in their approach. Hla and Kavi (2008) described the need and possible techniques for segmentation of Myanmar script. They used a combination of stored lists, suffix removal, morphological analysis and syllable-level n-grams to hypothesize valid words with an accuracy of about 99%. They built a list of 1216 stop words, 4550 syllables and 800,000 words from a variety of sources, including their own corpora. Tin and Mikami (2010) proposed the automatic syllable segmentation of Myanmar text using a finite state transducer, without using step-by-step heuristic rules and an annotated corpus. They proved that this approach could handle both the regular and irregular syllable structures of Myanmar with acceptable performance. Although they did not publish this syllable segmentation software, their segmentation of the Myanmar syllable was an important step.

3. Proposed Method

The overall process of the proposed method is shown in Figure 1.

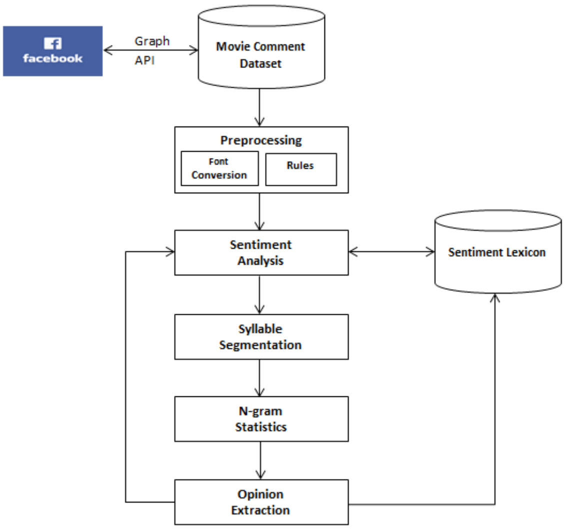


Figure 1: The proposed system

Firstly, an initial sentiment lexicon containing a small number of opinion words is manually created. Facebook comments concerning movies are retrieved and preprocessed. Next, each comment is classified as positive or negative using the initial lexicon. Then, an arbitrary n-gram of syllables is extracted using a publicly available syllable segmentation tool as candidates of the opinion words. Next, the opinion words in the movie domain are chosen from these candidates, based on the statistics of the syllable n-grams, and added to the sentiment lexicon. The procedures for the sentiment analysis of the comments and acquisition of the opinion words are repeated until no opinion word is obtained.

3.1 Data

Before applying the preprocessing step, it is common practice to collect data. Accurate data collection is important, as it helps to ensure the integrity of research. If existing data are not accurate, or do not provide enough information or the right kind of information, the study cannot be validated, misleading other researchers into pursuing fruitless avenues of investigation and distorting findings, resulting in wasted resources.

3.1.1 Collection of Facebook Movie Comments

In the last few years, Facebook has taken the world by storm and has become an important element in the field of communications. The Facebook Myanmar movie comment dataset is a corpus of movie comments used for the automatic acquisition of new opinion words. The original unprocessed Facebook comments used in this research were collected using the Graph API,[2] a programming tool designed to support greater access to conventions on the Facebook social media platform. Statistics for these data are shown in Table 1. In this table, the numbers of positive and negative comments are counted manually; note that the manually annotated polarity of the comment is not used for construction of the sentiment lexicon in the proposed method.

Type	Number
Facebook Movie Web Page Links	14
Posts	394
Comments	12,123
Syllables	92,167
Positive Comments	5,697
Negative Comments	458

Table 1: Statistics of data

3.1.2 Json to csv converter

All comments from movie posts are retrieved and the Json file of extracted comments is converted into a csv file using a converter[3] and then transformed to txt format, turning the separate movie files into a single file.

3.2 Preprocessing

Data preprocessing allows the production of a higher quality of text classification and a reduction in the computational complexity. Our preprocessing procedure includes the following steps.

3.2.1 Font Conversion

There is an issue with fonts in the Myanmar language; most users are very familiar with the Zawgyi font, and use this in Facebook comments, while most applications in the

[2] https://developers.facebook.com/tools-and-support/

[3] https://json-csv.com/

technology field accept only Unicode. Thus, in this research, the collected comments in Zawgyi are converted to Unicode using an online converter.[4]

3.2.2 Rules for Preprocessing

This is the conversion step from unstructured to structured data. Before starting the creation of n-grams, it is necessary to carry out several preprocessing and cleaning steps. One difficulty in the processing of most comments and documents is the presence of certain kinds of textual errors, such as spelling and grammatical errors. An analysis of data that have not been carefully screened can produce misleading results. We therefore investigate the influence of preprocessing rules on the quality of the system. To do this, the system performs the following preprocessing steps:

- Emoticons are replaced with their Myanmar words.

  For example: :) = ပြုံးသည် (smile) :(= မဲ့သည် (sad)
- Several useful English words and loan words are first converted to lower case and then translated to similar Myanmar words, for example: ‘good’ to ကောင်းသည် , ‘like’ to ကြိုက်သည် , ‘academy’ to အကယ်ဒမီ
- English acronyms are translated to similar Myanmar words. For example: ‘LOL’ to အရမ်းရယ်ရသည်, ‘WOW’ to အံ့အားသင့်သည်
- English-Myanmar combined words are translated to meaningful Myanmar words. For example: ‘Sပေါ’ , ‘$ပေါ’ to အရမ်းပေါ (so bad)
- Myanmar words with English pronunciation are translated to Myanmar words, for example: ဂွတ်တယ် (good in English) to ကောင်းတယ်
- Certain Myanmar synonym adverbs like အားကြီး, အကုန်, အသေ are replaced by အရမ်း (very).
- Repeated Myanmar syllables are replaced by three syllables. For example: ဒုန်းဒုန်းဒုန်း (encourage) , ခစ်ခစ်ခစ် (laugh)
- Punctuations and other non-Myanmar words in the comments are deleted, and white space is reduced.
- Words in comparative form are replaced with the basic form. Examples of such words are ပိုတော် (cleverer), အတော်ဆုံး (cleverest)

3.2.3 Syllable Segmentation

This refers to the ability to identify the components of a word, phrase, or sentence. Since Myanmar is a tonal language and has a syllabic writing system, the fundamental building blocks of the language are syllables. Almost every syllable has a meaning in the Myanmar language, and this can also be seen in the work of Hopple (2003). We use publicly available syllable segmentation software[5] to segment the syllables in Facebook comments and apply these segmented syllables in the calculation of n-gram statistics.

In Myanmar, unlike in English, word segmentation is not clearly denoted in sentences. For this type of language, the sentences are usually divided into a sequence of words in preprocessing. However, since Myanmar is an under-resourced language, no public tool for word segmentation is available. Therefore, in this study, the arbitrary n-grams of the syllables are extracted as candidates for the opinion words. Some of them are not genuine words, and irrelevant syllable n-grams are automatically discarded. The acquisition of Myanmar opinion words is more challenging than in other, more richly resourced languages.

3.3 Initial Lexicon

A sentiment lexicon is a list of positive and negative opinion words. Positive opinion words are used to express desired states, while negative opinion words are used to express undesired states. Most of the opinion words are not of one syllable, since a phrase such as “ဒီရုပ်ရှင်က မကောင်းဘူး” (The movie is not good) can be wrongly classified as a positive sentence, due to the syllable ကောင်း (good). In the proposed system, the initial sentiment lexicon contains one positive word “ကြိုက်” (like) and one negative word “ပေါ” (bad). These are the most common words in the comments, and Figures 2 and 3 show the method used to find these initial positive and negative seeds within the comments.

```
Input: Facebook comments
Output: a positive seed
1. Read all Facebook movie comments.
2. Find unigram syllable count, combine duplicate
   syllables and sort them in descending order.
3. Select highest 1% of total unigram count and remove
   stopwords from these.
4. Reduce the unigram count by the bigram count and
   negation (for example, subtract the count of “not
   good” from the count of “good”).
5. Sort unigram syllables again and manually select the
   highest frequency positive unigram syllable as an
   initial positive seed.
```

Figure 2: Algorithm for initial positive seed selection

```
Input: Facebook comments
Output: a negative seed
1. Read all Facebook movie comments.
2. Find unigram and bigram syllable counts, combine
   duplicate syllables and sort these in descending order.
3. Select highest 1% of total unigram count and total
   bigram count, and remove stopwords from these.
4. Manually select the highest frequency negative unigram
   or bigram syllable as an initial negative seed (this is
   often found in both unigram and bigram (“bad” and
   “not good”) in the comments)
```

Figure 3: Algorithm for initial negative seed selection

3.4 Sentiment Analysis of Facebook Comments

The classification of a text or a sentence according to its semantic orientation or polarity (positive, negative or neutral) can be performed by several methods including

[4] http://www.mcfmyanmar.org/myanmar-unicode-converter/

[5] http://myanmarnlpteam.blogspot.jp/2008/02/syllable-segmentation-software.html

machine learning, lexicon-based methods or hybrid methods. We use a lexicon-based approach to discover sentiments. Prior polarities are defined by the initial lexicon, and the polarity value of a comment is a comparison of the prior polarities of its sentiment words. If the positive (greater than zero) opinion count is greater than the negative count, it is classified as positive; otherwise, it is classified as negative. The algorithm for polarity classification of the comments is shown in Figure 4. After sentiment analysis, we obtain a set of positive and negative sentences, and these are used in the next step.

```
read a comment in dataset
while there is a comment in dataset do
    pcount ←0
    ncount ←0
    read an opinion word in lexicon
    while there is an opinion word in lexicon do
        if the comment contains positive opinion word
        then        pcount ←pcount+1
        if the comment contains negative opinion word
        then        ncount←ncount+1
    end while
    if(pcount>ncount) then
        classify the comment as positive
    else if (ncount>pcount) then
        classify the comment as negative
end while
```

Figure 4: Algorithm for polarity classification of comments

3.5 Opinion Word Extraction

Following data collection and preprocessing, opinion words are extracted from a set of positive and negative sentences, based on the statistics of syllable n-grams.

3.5.1 N-grams

In the fields of computational linguistics, an n-gram is a contiguous sequence of n items from a given sequence of text or speech. The items can be phonemes, syllables, letters, words or base pairs, according to the application. The n-grams are typically collected from a text or speech corpus. Statistics for the combination of letters (n-grams) are very useful, as this offers substantial savings in terms of human effort, and the use of n-gram statistics in opinion acquisition is therefore very attractive. Statistics for high-frequency words tend to be more reliable than those for low-frequency ones. The syllable n-gram representation of the Myanmar word 'အကယ်ဒမီ (academy)' is shown in Table 2.

n=1	n=2	n=3	n=4
အ (a)	အကယ် (aca)	အကယ်ဒ (acade)	အကယ်ဒမီ (academy)
ကယ် (ca)	ကယ်ဒ (cade)	ကယ်ဒမီ (cademy)	
ဒ (de)	ဒမီ (demy)		
မီ (my)			

Table 2: Syllable n-gram representation of the Myanmar word 'အကယ်ဒမီ (academy)'

3.5.2 Generation of N-gram Frequency Statistics

The generation of n-grams starts with the corpus of preprocessed Facebook comments (sentences). From this corpus, all relevant n-grams (1-, 2-, 3-, 4- and 5-grams) are created, and each n-gram is associated with its frequency, i.e. the number of times it occurs in the corpus. It is very difficult to calculate n-grams for a large value of n, due to the memory limitations of computers. Most of the Facebook comments have an average of 14 syllables per sentence; 5-grams and below are enough for movie opinions, and these were therefore calculated for the corpus.

no	1-grams		2-grams		3-grams		4-grams		5-grams	
	word	count	word	count	word	count	word	count	word	count
1	တယ်	3846	အ ရမ်း	1086	ကြည့် ချင် နေ	349	အ ကယ် ဒ မီ	229	အ ရမ်း ကြိုက် နေ တယ်	136
2	အ	3488	ပါ တယ်	891	အား ပေး နေ	344	ကြည့် ချင် နေ တယ်	145	အ ကယ် ဒ မီ ရ	108
3	ပါ	2890	အား ပေး	859	အ ရမ်း ကြိုက်	306	ကယ် ဒ မီ ရ	140	အ ရမ်း အား ပေး နေ	77
4	မ	2326	ကြည့် ချင်	779	ကယ် ဒ မီ	277	အား ပေး နေ ပါ	113	ကယ် ဒ မီ ရ ပါ	75
5	ကို	2043	ကြိုက် တယ်	596	အ ကယ် ဒ	235	ပေး နေ ပါ တယ်	110	ဒ မီ ရ ပါ စေ	58
6	ပေး	1429	ပေး ပါ	472	တင် ပေး ပါ	234	အ ရမ်းကြိုက် တယ်	103	အ မြဲ အား ပေး နေ	39
7	တာ	1383	တင် ပေး	415	သ ရုပ် ဆောင်	232	အား ပေး နေ တယ်	102	အား ပေး ချင် နေ တာ	37
8	ကြိုက်	1339	ပေး နေ	378	ဟိန်း ဝေ ယံ	176	အ မြဲ အား ပေး	101	အ ရမ်း ကြည့် ချင်တယ်	29
9	ကြည့်	1306	ရမ်း ကြိုက်	369	ကြည့် ချင် တယ်	175	ကို မြင့် မြတ် ကို	93	အ ကယ် ဒ မီ ရ	28
10	က	1246	မိုး စက်	355	အ ရမ်း ကောင်း	165	အား ပေး ပါတယ်	86	အား ပေး နေ ပါ တယ်	27
11	ရမ်း	1242	မြင့် မြတ်	352	ဒ မီ ရ	148	ဒ မီ ရ ပါ	81	ကယ် ဒ မီ ရ ပါ	27
12	ရ	1161	အ ကယ်	315	မိုး စက် မေ	145	မီ ရ ပါစေ	77	ဒ မီ ရ ပါ စေ	25
13	ကြ	1146	ချင် တယ်	315	အ ရမ်း ချစ်	140	တင် ပေး ပါလား	75	အ မြဲ အား ပေး နေ	25
14	နေ	1097	ကယ် ဒ	276	နေ ပါ တယ်	134	သ ရုပ် ဆောင် တာ	73	အ ရမ်း ကြည့် ချင် လိုက်	24
15	ကား	1071	ဒ မီ	249	ရ ပါ စေ	127	မြဲ အား ပေး နေ	62	ကြည့် ချင် တာ အ ရမ်း	24

Table 3: Occurrence of the top 15 opinion n-grams from positive comments

3.5.3 Comparing and Ranking N-Gram Frequencies

The scores of n-grams are sorted in descending order and checked as to whether the highly ranked n-grams are valid opinion words. We generate at most 5-grams (n=5) and the highest ranking n-grams are mostly bi-grams (n=2); as we move towards longer n-grams, we generally obtain fewer valid words. It should be noted that these observations apply mostly to shorter comments with an average of about 14 syllables per sentence, such as those from Facebook movie webpage links. If these sentences were longer, a shift from 5-grams to higher n-grams would be seen. The ranking of the top 15 n-gram frequencies for positive comments is shown in Table 3, and n-gram statistics for Myanmar syllables are shown in Figure 5.

3.5.4 Score for Opinion Word Extraction

N-grams of syllables are candidates for opinion words. We calculate ScorePOS(*w*), the score of the positive orientation of each possible word *w* (syllable n-gram). First, the probability of generating *w* in the sets of positive sentences and all sentences, Ppos(*w*) and Pall(*w*), are calculated using Equations (1) and (2). Then, ScorePOS(*w*) is defined using Equation (3), where fre_pos(*w*) represents the count of *w* in positive opinion sentences. We assign a higher score to words that appear relatively frequently in positive sentences, and give a lower score to more general words. Frequent words are also preferred as positive opinion words to be added into the sentiment lexicon. The score of the negative orientation, ScoreNEG(*w*), can be calculated similarly. Finally, the most highly ranked candidates are extracted as positive or negative opinion words.

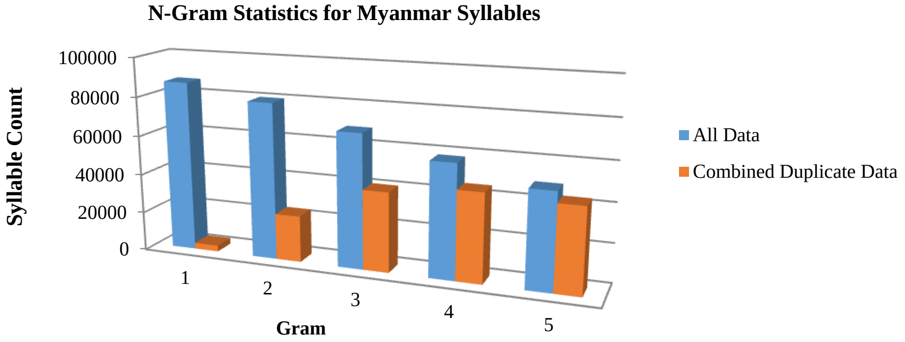


Figure 5: N-gram statistics for Myanmar syllables

$$Ppos(w) = \frac{count\ of\ w\ in\ positive\ opinionated\ sentences}{count\ of\ all\ n-grams\ in\ positive\ opinionated\ sentences} \quad (1)$$

$$Pall(w) = \frac{count\ of\ w\ in\ all\ sentences}{count\ of\ all\ n-grams\ in\ all\ sentences} \quad (2)$$

$$ScorePOS(w) = log\frac{Ppos\ (w)}{Pall\ (w)} * fre_{pos(w)} \quad (3)$$

3.5.5 Extraction of Opinion Words Using Bootstrapping Method

Starting from the initial lexicon with two words, syllable n-grams with scores greater than or equal to five are extracted as potential new opinion words[6]. In order to choose only meaningful words, the potential opinion word is added to the sentiment lexicon if it is a lemma or an inflectional form of a word in the Myanmar word list[7]. Unigrams are neglected to improve performance, since most of these words are meaningless. Furthermore, we do not add combinations of existing opinion words (for example, if an existing opinion word is “like”, its combinations of existing opinion words are “be like”, “like more”, “like most”, “like+stop word”, “other word+like”) and Myanmar stop words such as prepositions/postpositions, particles, inflections and conjunctions, which appear as suffixes of other words. When a word is assigned as both positive and negative, the difference of the polarity scores is considered. If its positive and negative scores are nearly equal, the word is not extracted as an opinion word; if the difference between ScorePOS(*w*) and ScoreNEG(*w*) is greater than three, the word is added as either a positive or negative word to the sentiment lexicon. The polarity of the added word is the same as the polarity of the higher score.

[6] The threshold of syllable n-grams is set to five by our intuition.

[7] https://github.com/kanyawtech/myanmar-karen-word-lists/blob/master/burmese-word-list.txt

4. Evaluation and Discussion

This section reports the results of the experiment conducted to evaluate our proposed method. After the first time with one opinion each for positive and negative, 11 and 10 new words were extracted as positive and negative opinion words, respectively. The program was run with 20 positive opinion words and 19 negative opinion words for the second time.

Our experiments showed that six and seven bootstrapping iterations were sufficient to extract 44 positive words and 35 negative words, respectively. These newly acquired words were evaluated in terms of precision. Precision is defined as the ratio between the number of correctly extracted opinion words and the total number of extracted words, as shown in Equation (4).

$$Precision = \frac{number\ of\ correctly\ extracted\ opinion\ words}{total\ number\ of\ extracted\ opinion\ words} \quad (4)$$

We manually checked the correctly extracted opinion words. The results of this evaluation of the lexicon obtained by bootstrapping are shown in Table 4. We found that the precisions of all positive and negative words were 86% and 89% respectively, which are relatively high.

Based on these bootstrap-based experiments, we found that most opinion words were syllable bi-grams and tri-grams, and their statistics act as a good scoring metric. Incorrect opinion words were retrieved; these were the names of actors that appeared many times in the comments, Myanmar adverbs and meaningful sub-words of other opinion words.

Round	Positive		Precision	Negative		Precision
	All extracted words	Correct words		All extracted words	Correct words	
1	11	8	73%	10	8	80%
2	31	26	84%	29	26	90%
3	41	35	85%	32	28	88%
4	43	37	86%	33	29	89%
5	44	38	86%	34	30	88%
6	-	-	-	35	31	89%
7				-	-	-

Table 4: Results of evaluation of the lexicon using bootstrapping

5. Conclusions and Future Work

In this paper, we propose a syllable-based n-gram approach for the automatic extraction of new opinion words from Facebook movie comments. These opinion words can be combined with existing opinion words to increase the accuracy of the opinion words and expand the lexicon. Opinion words are very important in sentiment analysis, and we believe that the proposed system can perform well in any domain to retrieve these words automatically.

A variety of steps can be taken to extend this work:

- We need to develop a larger corpus of movie documents and more data cleaning rules.
- We plan to handle spelling variations and to incorporate our method with a word segmentation tool.
- We intend to experiment further using a wider variety of Facebook comments, which we hope will give rise to more opinion words.
- We plan to perform statistical substring reduction (SSR) on the acquired n-gram statistics.
- We need to handle slang words.
- We need to handle implicit emotional comments, where the sentiment of an author is not literally written.
- We also intend to apply our approach to other domains.

6. Bibliographical References

Bravo-Marquez, F., Frank, E. and Pfahringer, B. (2015a). From unlabelled tweets to Twitter-specific opinion words. In Proceedings of the 38th International ACM SIGIR Conference on Research & Development in Information Retrieval, pp. 743-746.

Bravo-Marquez, F., Frank, E. and Pfahringer, B. (2015b). Positive, Negative, or Neutral: Learning an Expanded Opinion Lexicon from Emoticon-annotated Tweets. In Q. Yang & M. Wooldridge (Eds.), Proceedings of the 24th International Joint Conference on Artificial Intelligence, pp. 1229-1235), Buenos Aires, Argentina: AAAI Press.

Bravo-Marquez, F., Frank, E. and Pfahringer, B. (2016). Determining Word-Emotion Associations from Tweets by Multi-label Classification. IEEE/WIC/ACM International Conference on Web Intelligence (WI), Omaha, NE, 2016, pp. 536-539. doi: 10.1109/WI. 2016.0091

Htay, H. H. and Murthy. K. N. (2008). Myanmar Word Segmentation using Syllable Level Longest

Matching. In Proceedings of the 6th Workshop on Asian Language Resources, ACL ID I08-7006.

Nagao, M. and Mori, S. (1994). A New Method of N-gram Statistics for Large Number of n and Automatic Extraction of Words and Phrases from Large Text Data of Japanese. In Proceedings of the 15th Conference on Computational linguistics (COLING '94), Volume 1, pp. 611-615.

Souza, M., Vieira, R., Busetti, D., Chishman, R. and Alves, I. M. (2011). Construction of a Portuguese Opinion Lexicon from Multiple Resources. In Proceedings of the 8th Brazilian Symposium on Information and Human Language Technology, STIL, Mato Grosso.

Hopple, P. (2003). The structure of nominalization in Burmese, Ph.D. Thesis.

Remus, R., Quasthoff, U. and Heyer, G. (2010). SentiWS: A Publicly Available German-language Resource for Sentiment Analysis. In Proceedings of the 7th International Language Resources and Evaluation (LREC'10), pp. 1168-1171.

Hlaing T. H. and Mikami. Y. (2013). Automatic Syllable Segmentation of Myanmar Texts Using Finite State Transducer. The International Journal on Advances in ICT for Emerging Regions (ICTer), Volume 6, Number 2.

Maung, Z. M. and Mikami. Y. (2008). A Rule-based Syllable Segmentation of Myanmar Text. In Proceedings of the IJCNLP-08 Workshop on NLP for Less Privileged Languages, pp. 51-58, ACL ID I08-3010.