LREC 2018 Workshop

MLP–MomenT 2018:
The Second Workshop on Multi-Language Processing in a Globalising World
The First Workshop on Multilingualism at the intersection of Knowledge Bases and Machine Translation

PROCEEDINGS

Edited by

Jinhua Du, Mihael Arcan, Qun Liu, Hitoshi Isahara

ISBN: 979-10-95546-15-3
EAN: 9791095546153

12 May 2018

Proceedings of the LREC 2018 Workshop
"MLP–MomenT: The Second Workshop on Multi-Language Processing in a Globalising World and The First Workshop on Multilingualism at the intersection of Knowledge Bases and Machine Translation"

12 May 2018 – Miyazaki, Japan

Edited by Jinhua Du, Mihael Arcan, Qun Liu, Hitoshi Isahara

http://mlp.computing.dcu.ie/

Acknowledgments: This work is supported by the ADAPT Centre for Digital Content Technology, funded under the Science Foundation Ireland (SFI) Research Centres Programme (Grant 13/RC/2106) and by the Insight Centre for Data Analytics, funded by SFI under Grant Number SFI/12/RC/2289.

ADAPT
Engaging Content
Engaging People

Insight
Centre for Data Analytics

sfi
Science
Foundation
Ireland

Organising Committee

- Qun Liu, Dublin City University, Ireland
- Hitoshi Isahara, Toyohashi University of Technology, Japan
- Jinhua Du, Dublin City University, Ireland*
- Mihael Arcan, National University of Ireland Galway, Ireland*
- Tatjana Gornostaja, Tilde, Latvia, ELRA, BDVA
- Darja Fišer, University of Ljubljana, Jožef Stefan Institute, CLARIN ERIC
- Elena Montiel-Ponsoda, Universidad Politécnica de Madrid, Spain
- Chao-Hong Liu, Dublin City University, Ireland
- Joachim Wagner, Dublin City University, Ireland
- Mikel L. Forcada, Universitat d'Alacant, Spain
- Qi Zhang, Dublin City University, Ireland

*: Main editors and chairs of the Organising Committee

Programme Committee

- Guadalupe Aguado-de-Cea, Universidad Politécnica de Madrid, Spain
- Paul Buitelaar, National University of Ireland Galway, Ireland
- Philipp Cimiano, Bielefeld University, Germany
- Thatsanee Chaeronporn, Burapha University, Thailand
- Christian Chiarcos, Goethe-Universität, Germany
- Christopher Crowhurst, United Language Group, USA
- Beatrice Daille, University of Nantes, France
- Brian Davis, Maynooth University, Ireland
- Thierry Declerck, German Research Center for Artificial Intelligence (DFKI), Germany
- Mauro Dragoni, Fondazione Bruno Kessler (FBK), Italy
- Tomaž Erjavec, Jožef Stefan Institute, Slovenia
- Miquel Esplà-Gomis, Universitat d'Alacant, Spain
- Natalia Grabar, Université de Lille, France
- Jorge Gracia, University of Zaragoza, Spain
- Miloš Jakubíček, University in Brno / Lexical Computing Limited, Czech
- John Judge, Dublin City University, Ireland
- Kyoko Kanzaki, Toyohashi University of Technology, Japan
- Ilan Kernerman, K Dictionaries, Israel
- Simon Krek, Jožef Stefan Institute, Slovenia
- Els Lefever, Ghent University, Belgium
- Qing Ma, Ryukoku University, Japan
- Gudrun Magnusdottir, ESTeam AB, Sweden / Coreon GmbH, Germany
- John Philip McCrae, National University of Ireland Galway, Ireland
- Yohei Murakami, Ritsumeikan University, Japan

Table of Contents

Inspection of Multilingual Neural Machine Translation
Carlos Mullov, Jan Niehues, Alexander Waibel 1

Multilingual Semantic Networks for Data-driven Interlingua Seq2Seq Systems
Cristina España-Bonet, Josef van Genabith 8

Bianet: A Parallel News Corpus in Turkish, Kurdish and English
Duygu Ataman 14

Active Learning for Tibetan Named Entity Recognition based on CRF
Fei-Fei Liu, Zhi-Juan Wang 18

Phonetically Based Extraction of Japanese Synonyms from Rakuten Ichiba's Item Titles
Ohnmar Htun, Koji Murakami, Yu Hirate 23

Terminology Translation Accuracy in Phrase-Based versus Neural MT: An Evaluation for the English-Slovene Language Pair
Špela Vintar 34

Low-Resource Neural Machine Translation with Transfer Learning
Tao Feng, Miao Li, Lei Chen 30

A Semi-supervised Learning Approach for Person Name Recognition in Tibetan
Zhijuan Wang, Fuxian Li 38

Inspection of Multilingual Neural Machine Translation

Carlos Mullov, Jan Niehues, Alexander Waibel
Institute for Anthropomatics and Robotics
KIT - Karlsruhe Institute of Technology, Germany
uvdxd@student.kit.edu, jan.niehues@kit.edu, alexander.waibel@kit.edu

Abstract

In this paper we inspect the intermediate sentence representation in the multilingual attention-based NMT system proposed by Ha et al. (2016). We ask the question of how well the NMT system learns a shared representation across multiple languages, as such a shared representation is an important prerequisite for zero-shot translation. To this end we examine whether the sentence representation is independent of the individual languages involved in translation. Having found the sentence representation in our multilingual NMT system to be language dependent, we further inspect the sentence representation for the cause of this dependence. We isolated the language dependent features, and found present a linear correlation between the sentence representation and its source language. Using these isolated features, we describe a method to manipulate these features, and provide a way to eliminate the language specific differences between the sentence representations. This could potentially help to remove noise, which is particularly harmful for zero-shot translation.

Keywords: multilingual, neural machine translation, neural representation analysis

1. Introduction

Since the introduction of neural machine translation (NMT) in recent years, the field of machine translation has made significant progress. However, current NMT systems require large amounts of data for training, while there is a severe lack of data for most language pairs. Therefore, for such language pairs workarounds such as translation using a pivot language are needed. To address this problem Ha et al. (2016) have proposed to extend the originally bilingual attention-based NMT system (Bahdanau et al., 2014) to multilingual translation. Without changes to the network architecture, a single NMT system jointly learns to translate from multiple languages to multiple languages. The attention-based NMT system (Bahdanau et al., 2014) is an encoder-decoder architecture with an attention layer in between the encoder and the decoder, and in the translation process it produces an intermediate sentence representation, the so called *context vectors*. By extracting common semantics across multiple languages, the multilingual NMT system is expected to learn in its sentence representation a shared representation across these languages, as a result significantly reducing the amount of training data needed for each individual language pair, and in the extreme case even enabling zero-shot translation. This shared representation for languages closely resembles an open-domain interlingua, which is a linguistic concept that has often been considered impossible to achieve. Ha et al. (2016) have shown in their experiments, that while zero-shot translation is possible with their approach, there is a significant drop in translation quality.

With the aim to find clues as to what leads to this drop in quality, in this paper we examine how well the NMT system proposed by Ha et al. (2016) learns a shared representation by measuring how independent of the individual languages the sentence representation in the NMT system is. Furthermore, having found in our experiments that the sentence representation is language dependent, we explore the cause for this dependence. We discover a linear correlation between the sentence representation and the individual languages, and find a method to manipulate this correlation in a stable manner. Finally, we demonstrate that through this manipulation we successfully eliminate language dependent linear differences in the sentence representation. This manipulation of the sentence representation could potentially provide a way to manipulate the sentence representation in the process of translation, and effectively reduce noise for zero-shot translation.

2. Related Work

2.1. Multilingual Attention-Based Translation

Based on the attention-based encoder-decoder architecture described by Bahdanau et al. (2014), Ha et al. (2016) have proposed an approach to extend this architecture to multilingual translation. The idea is to train the NMT system using a unified vocabulary and training corpus across all languages, while making no modifications to the architecture. For this, Ha et al. (2016) describe two techniques in the form of input pre-processing steps:

- **Language-specific Coding** words of different languages are distinguished through language codes. This, for example, can look like the following: *bank* → *@**en**@bank*
- **Target Enforcing** A symbol indicating the desired target language of the translation is added to the beginning and at the end of the sentence.

Shared Embedding The approach of using shared vocabularies across multiple languages also results in a shared embedding space. As Ha et al. (2016) have shown in their experiments, the NMT system learns to correlate words of different languages in this shared embedding space in such a way, that words with similar meanings end up closer to each other.

The goals of this approach are to

- improve translation quality for individual languages, by letting the NMT system learn common semantics across languages, thus helping the system to better generalize

- improve translation quality for language pairs, for which parallel training data are scarce, and in the extreme case even allowing for zero-shot translation, by letting the NMT system find a representation of the sentences, which abstracts from the individual languages

- reduce the amount of translation systems needed for translation between n languages from $n\,(n-1)$ to a single one, thus reducing the amount of training time and the amount of parameters

2.2. Inspection of Neural Sentence Representations

Prior to the introduction of attention to the encoder-decoder architecture Cho et al. (2014), Sutskever et al. (2014) and Shi et al. (2016) among others have inspected the encoder sentence representation. The former two have explored the ability of the encoder to represent sentences at the level of their meaning by comparing the relative positions of sentences close to each other in terms of meaning. In their visualization of selected few sentences by means of a 2-dimensional PCA projection Sutskever et al. (2014) show a discernible additive relation between sentence representations, closely resembling the relation between words in word embeddings. This indicates that the encoder in their NMT system does indeed have the capability of abstracting from language and representing the translated sentence on a semantic level.

Shi et al. (2016) have inspected the sentence representation on a syntactic level and have found that the encoder implicitly learns to store information about the source sentence syntax in the sentence representation.

This sentence level representation, which Cho et al. (2014) call *summary* is the equivalent to the context vectors in the attention-based model, with the difference being that context vectors represent only the part of the sentence it puts attention to.

2.3. Inspection of Context Vectors in the GNMT System

Based on the same principle as (Ha et al., 2016) multilingual NMT system, Johnson et al. (2016) have proposed a multilingual attention-based encoder-decoder NMT system. In the course of their experiments Johnson et al. (2016) have inspected the intermediate sentence representation of the translated sentences in their NMT system in respect to its resemblance of an interlingua representation. This sentence representation they call the *attention vectors*, and is equivalent to what we call context vectors in this paper. Using a t-SNE projection into three-dimensional space, Johnson et al. (2016) observe that attention vectors for semantically identical sentences form clusters. Thus they visually confirm, that their NMT system learns to organize sentence representation by their meaning, which they call "early evidence of shared semantic representations (interlingua) between languages". Furthermore Johnson et al. (2016) have found a correlation between the translation quality for these semantically identical sentences of different languages and the similarity between the attention vectors for these sentences.

2.4. Generative Adversarial Networks

Generative Adversarial Networks (Goodfellow et al., 2014) are a type of neural network, used in an approach for training generative models in an unsupervised fashion. It consists of a generative network G, which tries to generate data and a discriminative network D, which tries to differentiate between data generated by G and the training data. G is then trained to "trick D into thinking" that the data generated by G originates from the training data by maximizing the error for D. In this adversarial manner G is trained to produce data which is indistinguishable from the actual training data.

The approach of letting a discriminating network D classify the output of another network G is similar to the procedure used in this paper: we build a discriminator D on top of the attention mechanism of a NMT system G, while ideally looking for D to fail in its classification task. Unlike with the approach with GANs however, we do not take the next step of adjusting G to maximize the error for D. We describe the potential future work on this matter in Section 5.1..

3. Inspection of Context Vectors

Prior to the introduction of attention to encoder-decoder NMT systems (Cho et al., 2014), a source sentence read by the encoder was encoded into a fixed length vector. The decoder then generated the target sentence having only seen this fixed length vector, forcing the encoder to find a meaningful sentence representation containing all the semantic information in the source sentence. With attention this meaningful representation has moved from this single fixed length vector to a set of multiple vectors, the so called context vectors; the principle however stays the same. The addition of multiple languages to the source and target side of the encoder and decoder as proposed by Ha et al. (2016) increases the problem complexity while keeping the amount of parameters constant, thus compelling the network to generalize by using common semantics between languages. Under such circumstances the NMT system would ideally learn a purely semantic representation of sentences, while abstracting from the individual source and target languages. This principle of translating a sentence into its language-independent meaning is known in linguistics as an interlingua representation, and the idea has been known for many centuries. As the context vectors are strongly reminiscent of such an interlingua representation, this begs the question of how close to an interlingua it is. In other words, we want to know how well the NMT system learns a shared semantic representation of sentences across multiple languages.

One criterion for evaluating how well the NMT system learns this shared representation is to look at the independence of the sentence representation – the context vectors – from the individual languages. In Section 3.1. we describe how we measure this degree of independence for the context vectors in the proposed NMT system.

Based on our experiments we have found the context vectors in our NMT system to be language dependent. In Section 3.2. we describe how we – while exploring the cause for the dependence – isolated the language dependent features in the context vectors. Furthermore we describe how we use the result to manipulate the context vectors. Finally we describe how we confirm that using this manipulation we can eliminate the linear language specific differences between context vectors. This could potentially be applied in zero-shot translation in order to change context vectors of a language pair which the NMT system never saw during the training to take on the form of context vectors which the NMT system saw during the training, effectively removing noise in the process of translation.

3.1. Measuring Context Vector Independence

We consider the independence of the context vectors from the individual languages to be a good indicator for how well the NMT system learns the shared representation. This is because, assuming that the NMT system perfectly learns a shared representation, then sentences of different languages would arrive at the same representation and would thus be indistinguishable in terms of the languages involved. Considering the fact that showing the independence of the context vectors from the source and target language would require a formal proof, we approach the problem by studying the dependency, which, if present, can be discovered through experimental means. Given a context vector, we try to identify the language pair it was generated from, and declare the dependence in the case of success. As this problem can be formulated as finding a correlation between a vector $c \in \mathbb{R}^n$ and one language from a set of candidates $\{l_1, \ldots, l_k\}$, this calls for classification. Given the recent success of neural networks in discriminative tasks with high dimensional input, we approach this particular classification problem with classification via neural networks.

Neural Classification Due to the nature of our input we believe a simple feed-forward neural network (FFNN) with fully connected layers to be the most appropriate type of network as the basis for the classifier. Using supervised learning, the classifier is trained to predict the correct source-target language pair, by providing context vectors as input and their respective true source-target language pair as the label. Starting with the simplest approach we will first attempt linear classification using a network without any hidden layers. We call this type of network a *single layer perceptron* (SLP), and this type of classification *linear classification*. The capability of such a network to successfully detect the presence of a dependence would suggest a linear separability of the context vectors, allowing the network to partition the feature space into relevant classes. After the linear classification we will attempt a classification using an MLP-classifier with one or more hidden layers to look for nonlinear features.

The labels will be encoded as concatenation of two one-hot vectors, the first vector encoding the source language and the second vector encoding the target language. The classifiers will then be trained using a softmax-layer as the output layer and cross entropy error between the first and second half of the network output and labels:

$$\begin{aligned} o_s &= \mathrm{softmax}(D(x)_{1\ldots5}) & o_t &= \mathrm{softmax}(D(x)_{6\ldots10}) \\ t_s &= t_{1\ldots5} & t_t &= t_{6\ldots10} \\ E_s &= H(o_s, t_s) & E_t &= H(o_t, t_t) \end{aligned}$$

$$E = \frac{E_s + E_t}{2}$$

for the classifier D and the input-label pair (x, t). The network is then trained to minimize E using *adam* (Kingma and Ba, 2014) as optimizer. The classifier predicts the language pair using the $argmax$ of the output first and second half:

$$(L_s, L_t) = (argmax(D(x)_{1\ldots5}), argmax(D(x)_{6\ldots10}))$$

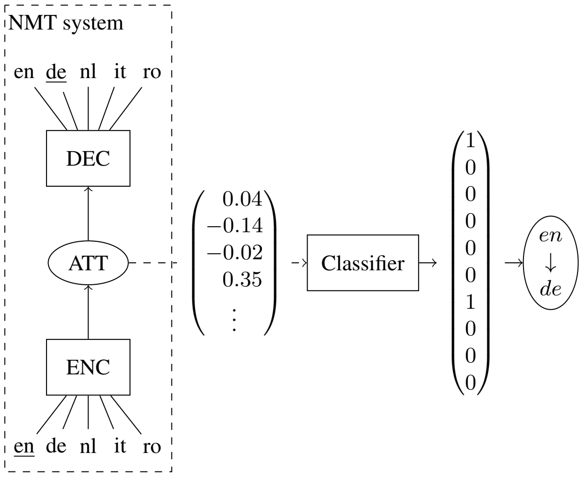

Figure 1: Schematic description of the procedure we use to test for the dependency of the context vectors. During translation the context vectors are extracted from the attention module, and fed to the classifier. The classifier output is a "two-hot" vector representing the predicted source and target language in the its first and second half (every language is assigned a fixed dimension, e.g. English in dimension 0,. . .).

3.2. Investigating Linear Relation of Context Vectors

The results of our experiments have shown that a linear classifier is capable of correctly classifying the context vectors. As described in Section 3.1., this suggests the linear separability of the context vectors. Suspecting the existence of a linear translation, such that a context vector of one language can be obtained when applied to a context vector of another language (similar to the additive relation between words with word embeddings), we decided to further investigate this matter.

Taking context vectors x with s_1 as their source language and t_1 as their target language, and another language s_2 we will try to find a vector b, such that $x + b$ is recognized as a context vector with s_2 as its source language. In order to find such a vector b for the context vector x we will need a comparable context vector x' which has s_2 as its source language. To obtain this counterpart x' for x, we need to ensure that two matching source sentences f_1 and f_2 in our parallel corpus with s_1 and s_2 as their respective language, are both translated into the same target sentence in the language t_1, allowing for the direct comparison of the generated context vectors. To this end we will adjust the NMT system sampling mechanism to accept a reference target sentence $e = (e_1, \dots, e_m)$ and use e_{t-1} as the input for the decoding step t, instead of the previously generated target word y_{t-1}. With this method we will obtain a pair of context vectors (x_t, x'_t) for each target word e_t, which can then be used as an input and its label in the training of b.

Using a training set with German-English context vectors for the input, and matching Dutch-English context vectors as labels, we will again use gradient descent to train a translation by randomly initializing a vector b and then minimizing the *summed squared error*

$$sse(x + b, t) = \frac{1}{2}\sum_i (x_i + b_i - t_i)^2$$

for each input-label pair (x, t).

The resulting translated vectors $x + b$ will then be fed to the previously trained classifier, in order to see whether they are recognized as Dutch-English context vectors. Furthermore, in order to see how this translation affects context vectors of different target languages, we will apply this translation b to German-Italian context vectors. We expect the resulting vectors to be classified as Dutch-Italian context vectors.

Eliminating Language Specific Differences After finding the linear translation b_{AB} which translates from the original source language A to the new source language B, we can eliminate the source language specific differences for context vectors c and c' with A and B as their respective source language, by translating c to $c+b_{AB}$. To confirm that the new context vectors are indeed language independent (at least linearly), we can again train a classifier as described in Section 3.1. using the modified context vectors. To this end, we (as illustrated in figure 2) take context vectors of different language pairs and translate each of them to English-German context vectors. We then train classifiers to predict the original language pair for these translated context vectors to see whether the classifiers are still able to differentiate between context vectors of different languages.

4. Evaluation

4.1. Multilingual Translation Models

In order to inspect the context vectors we have trained translation models as described by Ha et al. (2016).

Training Data For training we have used the multilingual WIT^3 (Cettolo et al., 2012) training corpus, which provides high quality multilingual translations. This corpus consists of transcriptions of 200,000 English sentences from TED Talks, and their translations into German, Dutch, Italian and Romanian. We have trained the NMT system using every possible combination of source and target languages, except for language pairs where the source and target language are the same. This gives us 20 language pairs, and therefore 20 valid classes of context vectors and a total of 4,347,886 sentence pairs. For the purpose of evaluation we have used another development dataset set consisting of 900 sentences from TED Talks involving the same five languages.

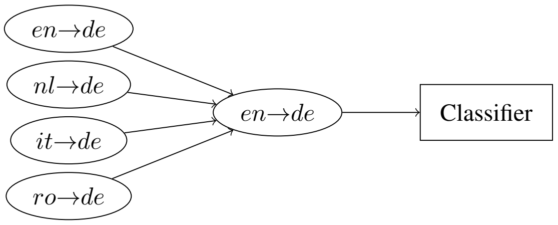

Figure 2: To see whether we can eliminate language specific differences between context vectors, we train classifiers to differentiate between translated context vectors.

Translation System We have trained the translation models using Nematus (Sennrich et al., 2017), which provides an attention-based encoder-decoder NMT system as proposed by Bahdanau et al. (2014). Nematus uses a special type of RNN, which Sennrich et al. (2017) call *conditional GRU with attention* in its decoder. As alignment model Nematus uses a feed-forward *tanh*-layer, which is jointly trained together with the rest of the system. We have used subword translation units.

In order to achieve multilinguality as described by Ha et al. (2016), all 20 parallel corpora have been merged into one multilingual parallel corpus through the concatenation of the source and target texts in the same respective order. The source and target vocabularies have been built from the merged corpus after applying every pre-processing step. The pre-processing steps include:

1. tokenization
2. true casing
3. byte pair encoding
4. language specific encoding
5. target enforcing

As we assume, that a smaller hidden layer size will force the network to abstract from the source language even more, we have trained models with different sizes of hidden layers in order to test whether this holds true. Furthermore, in order to compare how classification results evolve with the ongoing training of the translation system, we also evaluate one particular model at different checkpoints in the course of training.

Model Parameters For the evaluation task we have trained three translation models. We trained all models using the same setup and network parameters, differing only in the amount of training time and the hidden layer sizes, which is 1024 for the first model, and 512 for the second and third models. We used English, German, Dutch, Italian and Romanian as source as well as target languages, whereby words were pre-processed into subword units with BPE[1], using a BPE merging parameter of 39,500 trained on the merged corpus before applying language specific encoding, resulting in a total vocabulary size of 88,000 words. We used a maximum target sentence length of 50 and a word embedding size of 500.

Training We have trained all models using a batch size of 40, *adam* as optimizer and dropout training (Srivastava et al., 2014), with a dropout ratio of 0.2 for the embedding layers and hidden layers and 0.1 for the source and target layers. Since we use a concatenation of the bidirectional encoder forward and backward hidden states as annotation vectors, this resulted in context vectors of size 2048 for the first model and 1024 for the second and third models.
The first model, with the hidden layer size of 1024 was trained for 110,000 iterations, until coming to an early stop. This resulted in a final BLEU score of 11.94. Another model, with a hidden layer size of 512 was trained for 160,000 iterations, resulting in the second model. Training the second model for another 100,000 iterations, resulted in the third model after a total of 260,000 iterations. These models achieved BLEU scores of 11.07 after 160,000 iterations and 14.94 after 260,000 iterations (see Table 1).
All translations used for calculating BLEU scores were generated using beam search decoding, with beam size 5.

trg src	en	de	nl	it	ro	avg
en		17.82	16.17	17.27	14.75	16.50
de	23.21		13.84	12.07	9.65	14.69
nl	20.42	12.46		12.11	9.44	13.61
it	22.84	12.12	12.35		11.16	14.62
ro	22.85	11.87	12.23	14.10		15.26
avg	22.33	13.56	13.64	13.88	11.25	14.94

Table 1: BLEU scores for the third NMT model (hidden layer size 512 and 260,000 training iterations). The distribution of scores for specific language pairs is also representative for the other trained models.

4.2. Classification of Language Pairs

Using the development dataset as translation source, we generated and extracted the context vectors and the correct language pairs from *Nematus* using the previously trained models. This resulted in 459,878 context vectors for the first model, 457,054 vectors for the second model, and 444,570 vectors for the third model, with overall 20 classes. The number of context vectors matches the number of translation symbols in the generated target sentence, and thus differs for each translation model, since they produce different translations. We merged the data from each class into one sequence by alternating between single sentences of each class, ensuring that each mini batch of size 1000 to contained samples of all 20 classes, considering the sentence maximum length of 50.
Using the *TensorFlow* (Abadi et al., 2016) low level API we built for each model SLP-classifiers, and MLP-classifiers with one hidden layer containing 64 hidden neurons. As the hidden layer activation function we use *ReLU*. All classification accuracies were calculated as ratio of correctly predicted language pairs, using 25% of the context vectors as the validation set.

Results As illustrated in Table 2 all classifiers have achieved significantly high classification accuracies, with linear classification achieving 86-96% correct classification rates after 50 epochs of training, and slightly higher rates for their nonlinear counterparts. These values are also representative for the classification rates of the source languages alone, as the target languages have been correctly classified with near 100% accuracy by all the classifiers.

NMT model	linear	MLP
first	95.75%	96.09%
second	85.93%	89.94%
third	91.93%	94.29%

Table 2: Classification rates for the linear classifiers and the MLP-classifiers after 50 epochs of training. *NMT model* refers to the model which was used for generating the context vectors.

These results strongly suggest that the extracted context vectors are not independent of the source language. The high classification rates of the linear classifiers further suggest a linear relation between context vectors of different languages. In view of the fact, that the classification rates for the linear classifier increase with ongoing training of the NMT system, it is apparent that the linear features which the classifier makes use of become more distinctive with the progression of the training.
The confusion matrix (see Table 3) shows a discernible correlation between the language specific BLEU scores and classification errors for that language, Romanian being the language with lowest BLEU scores, as well as with most classification errors. Furthermore, there is also a noticeable correlation between language similarity and the classifiers tendency to confuse them with each other, as can be seen with Dutch being commonly misclassified as German and

label pred	en	de	nl	it	ro
en	22398	25	21	15	42
de	32	21818	129	19	24
nl	24	326	24784	43	89
it	171	46	74	19728	2490
ro	85	54	94	628	20841

Table 3: The Confusion matrix shows a comparison between predicted source languages and labels

[1] for BPE we used the script from the subword-nmt repository (`https://github.com/rsennrich/subword-nmt`). For tokenization and true casing we used the scripts provided by the Moses framework (`http://www.statmt.org/moses/`)

source language	original	translation
en	0	0
de	1268	133
nl	100	1246
it	32	16
ro	13	18

Table 4: Comparison of predicted source language for German to English context vectors, after training and applying translation of source language to Dutch

source language	original	translation
en	4	1
de	1233	172
nl	18	1081
it	0	0
ro	2	3

Table 5: Comparison of predicted source language after applying the same translation to German to Italian context vectors

Romanian misclassified as Italian.

For the classification with the MLP we can observe slight increases in classification rates.

4.3. Linear Relation between Context Vectors

To investigate the supposed linear relation between the context vectors of different languages, we successfully trained a translation as described in Section 3.2.. For this we first modified *Nematus* to accept a reference target sentence for translating a source sentence.

Using the German and Dutch translation of the development dataset as translation source and the English translation as the reference, we generated a training set with the German-English context vectors as the input and the Dutch-English context vectors as the labels, using the third translation model. Using *adam* as optimizer we then trained a vector b, by minimizing the *summed squared error*. The training of such a translation for 20 epochs resulted in a vector b with a norm of 0.710 and a mean distance of 6.912 between the translations and their labels, the mean distance between untranslated German-English vectors and their Dutch-English counterparts being 6.939 (see Figure 3).

Applying this trained translation to a validation set unseen in training, we further classified the originally German-English context vectors with the help of the previously trained linear classifier. As seen in Table 4, 88% of the translated vectors were classified as Dutch-English. Furthermore, the application of this translation to German-Italian context vectors, resulted in 86% of these to be classified as Dutch-Italian (see Table 5). Applying this same procedure for all 180 valid four-tuples of languages[2], from a total of 752,841 context vectors for 90.9% the source language was successfully translated as intended, while not affecting the predicted target language for these context vectors.

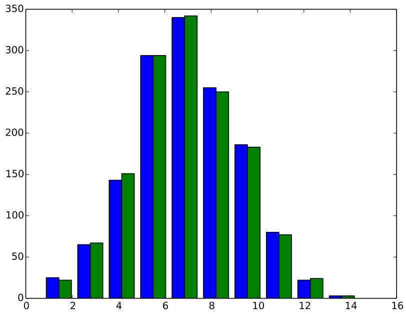

Figure 3: The distribution of distances between original (blue) and translated (green) German to English context vectors to their correspondent Dutch to English context vectors shows, that the translation has negligible impact on the distances.

4.3.1. Eliminating Linear Dependence

As described in Section 3.2. in order to confirm that the found linear translations can be used to eliminate the linear dependence of the context vectors, we have retrained our classifiers using modified context vectors. For the training set we took all the context vectors with German as their target language, and translated them to English as the new source language. Analogous to the procedure used in Section 3.1. we then trained classifiers to predict the original target language. For the linear classifiers this resulted in classification rates around 25% for 4 classes, thus failing at the classification task. This indicates that we successfully eliminated the linear dependence on the source language.

The MLP-classifiers achieved classification rates of up to 83.3%, showing the presence of purely non-linear language dependent features.

5. Conclusion

In this paper we have explored the ability of the multilingual NMT system proposed by Ha et al. (2016) to produce a shared representation across multiple languages. To this end we have inspected the intermediate sentence representation of the NMT system, the context vectors. We took as criterion for how well the NMT system learns a shared representation the degree of independence of the context vectors from the source and target languages involved in translation. In order to measure the dependence, we have trained classifiers based on feed-forward neural networks which, given a context vector, predict the language pair involved in its generation. Using the context vectors generated by our trained multilingual NMT systems, our linear

[2] For the original source language s, the translated source language s', the target language used in training the translation t, and the target language t' of the context vectors which the trained translation was applied to, a four-tuple is considered valid if s, s', t are pairwise different and s, s', t' are pairwise different, resulting in 60 different translations, which are each applied to context vectors of 3 different target languages.

classifiers have achieved rates of correct classification of up to 95.75% for 25 possible classes. This suggests that our NMT system does not successfully produce a shared representation.

Having explored the underlying cause of success in classification, we have found present a linear relation in Euclidean space between context vectors of different languages. More precisely, for a pair of languages (A, B) we have found a vector b_{AB}, such that for a context vector c with A as its source language, $c + b_{AB}$ is classified as having B as the source language in 90.9% of the cases. This translation of the source language does not affect the target language. We have found this translation to have negligible impact on the distance between the context vectors, which leads us to the belief that these language dependent differences in context vectors are merely noise, and particularly harmful for zero-shot translation. Finally we have demonstrated that our linear classifiers, which we trained on context vectors with modified source language fail in their task to classify the original source language. This shows that we can use translations found to effectively eliminate the language specific linear differences between context vectors.

5.1. Future Work

Adversarial Training of NMT System As described in Section 2.4., the approach used in this paper is similar to the first stage in the procedure to train generative models with GANs. As GANs have shown great success, the remaining steps of this procedure could be applied to the training of the NMT system as well. By training the NMT system G to produce context vectors for which a discriminating network D is unable to predict the correct language pair in an adversarial manner, the NMT system could learn to produce indistinguishable context vectors. The NMT system would then be alternatingly be trained in supervised learning and adversarial unsupervised learning, potentially learning a language independent representation.

Zero-Shot Translation The linear translation which we have found to be present between the context vectors could be potentially applied in order to improve zero-shot translation. Zero-shot translation is the task of producing translations for a language pair which the NMT system never saw during the training. For a language pair (A, C), which the NMT system never saw during the training, and a language pair (B, C), which the NMT system saw during training an attempt at improving translation quality for the unseen language pair could be made by translating the context vectors c_{AC} to $c_{BC} = c_{AC} + b_{AB}$ for the previously described translation b_{AB}. This produces context vectors for a language pair which the NMT system is more familiar with.

6. Bibliographical References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I. J., Harp, A., Irving, G., Isard, M., Jia, Y., Józefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D. G., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P. A., Vanhoucke, V., Vasudevan, V., Viégas, F. B., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *CoRR*, abs/1603.04467.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Cho, K., van Merrienboer, B., Gülçehre, Ç., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative Adversarial Networks. *ArXiv e-prints*, June.

Ha, T., Niehues, J., and Waibel, A. H. (2016). Toward multilingual neural machine translation with universal encoder and decoder. *CoRR*, abs/1611.04798.

Johnson, M., Schuster, M., Le, Q. V., Krikun, M., Wu, Y., Chen, Z., Thorat, N., Viégas, F. B., Wattenberg, M., Corrado, G., Hughes, M., and Dean, J. (2016). Google's multilingual neural machine translation system: Enabling zero-shot translation. *CoRR*, abs/1611.04558.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Sennrich, R., Firat, O., Cho, K., Birch, A., Haddow, B., Hitschler, J., Junczys-Dowmunt, M., Läubli, S., Barone, A. V. M., Mokry, J., and Nadejde, M. (2017). Nematus: a toolkit for neural machine translation. *CoRR*, abs/1703.04357.

Shi, X., Padhi, I., and Knight, K. (2016). Does string-based neural mt learn source syntax? In *Proceedings of EMNLP 2016*, pages 1526–1534, 01.

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215.

7. Language Resource References

Mauro Cettolo and Christian Girardi and Marcello Federico. (2012). *WIT^3: Web Inventory of Transcribed and Translated Talks*.

Multilingual Semantic Networks for Data-driven Interlingua Seq2Seq Systems

Cristina España-Bonet and Josef van Genabith
Universität des Saarlandes and Deutsche Forschungszentrum für Künstliche Intelligenz (DFKI)
Saarbrücken, Germany
{cristinae, Josef.Van_Genabith}@dfki.de

Abstract

Neural machine translation systems are state-of-the-art for most language pairs despite the fact that they are relatively recent and that because of this there is likely room for even further improvements. Here, we explore whether, and if so, to what extent, semantic networks can help improve NMT. In particular, we (*i*) study the contribution of the nodes of the semantic network, *synsets*, as factors in multilingual neural translation engines. We show that they improve a state-of-the-art baseline and that they facilitate the translation from languages that have not been seen at all in training (beyond zero-shot translation). Taking this idea to an extreme, we (*ii*) use synsets as the basic unit to encode the input and turn the source language into a data-driven interlingual language. This transformation boosts the performance of the neural system for unseen languages achieving an improvement of 4.9/6.3 and 8.2/8.7 points of BLEU/METEOR for $fr2en$ and $es2en$ respectively when neither corpora in fr or es has been used. In (*i*), the enhancement comes about because cross-language synsets help to cluster words by semantics irrespective of their language and to map the unknown words of a new language into the multilingual clusters. In (*ii*), because with the data-driven interlingua there is no *unknown* language if it is covered by the semantic network. However, non-content words are not represented in the semantic network, and a higher level of abstraction is still needed in order to go a step further and train these systems with only monolingual corpora for example.

Keywords: Multilingual Neural Machine Translation, Semantic Networks, BabelNet, Interlinguality

1. Introduction

The concept of semantic network was introduced by R.H. Richens in 1956 in relation to interlingual machine translation (IMT) (Richens, 1956). He defined a *semantic net* of *naked ideas* as what is left after removing the structural particularities of the base language. The elements of such a net represented things, qualities or relations. From 50 semantic primitives, Richens created the first semantic network, Nude, which was used for IMT. Modern semantic networks are usually implemented as semantic graphs, that are networks that represent semantic relationships between concepts where concepts are the vertices of the graph and edges represent semantic relations between them. Semantic networks have multiple uses. To date, machine translation is not among the most common ones.

A reason is that an interlingua representation in an open domain is difficult to achieve, and data-driven MT systems clearly outperform IMT for open-domain MT. Neural machine translation systems (NMT) are currently the state of the art for most language pairs (Bojar et al., 2017). Despite the success of this kind of architecture, it suffers from the same problem as other data-based translation systems: large amounts of parallel data must be available. To overcome this limitation, Artetxe et al. (2017) and Lample et al. (2017) introduce two unsupervised NMT methods that need only monolingual data but, up to now, they are far from the performance of seq2seq systems trained on bilingual corpora.

In this work, we investigate how a multilingual semantic network can be used for improving neural machine translation in general but specially for language pairs where not enough parallel data is available. We show how the inclusion of interlingual labels or synsets is beneficial in multilingual NMT (ML-NMT) systems and how they even allow beyond-zero-shot translation; that is, translation from languages that have not been seen in training. On the other hand, we explore a modern version of IMT, where the source text is codified into synsets and PoS tags and the translation into another natural language is learned by a seq2seq network.

Multilingual semantic networks have been used for machine translation mainly in statistical machine translation to deal with named entities and out-of-vocabulary words (Du et al., 2016; Srivastava et al., 2017). These issues are even more relevant in NMT because of the limited vocabulary that can be used to train the systems. However, the insights of seq2seq systems such as the difficulty to copy strings from the source into the target, make the integration a particular challenge.

The rest of the paper is organised as follows. Section 2. introduces BabelNet, the semantic network used for our experiments. Section 3. describes the NMT architecture and how the semantic information is included. Next, Section 4. describes the experiments and Section 5. analyses the results. Finally, Section 6. summarises and draws conclusions.

2. BabelNet

BabelNet (Navigli and Ponzetto, 2012) is a multilingual semantic network connecting concepts and named entities via *Babel synsets*. With 6 millions concepts and almost 8 millions named entities, the network covers 746 million word senses in 271 languages. This long list of languages, from Abkhazian to Zulu, includes many languages for which it is difficult to obtain parallel corpora.

Most of the concepts and named entities in BabelNet come from (Open Multilingual) WordNet, Wikipedia, Wikidata, Wiktionary and OmegaWiki. A synset groups these elements in different languages and treats them as synonyms in a language-independent way. The network also includes

	BabelNet				TED corpus	
Language (iso code)	Lemmas	Synsets	Senses	Synonym/Synset	Synsets	Coverage (%)
English (en)	11,769,205	6,667,855	17,265,977	2.59	28,445	27.25
French (fr)	5,301,989	4,141,338	7,145,031	1.73	–	–
German (de)	5,109,948	4,039,816	6,864,767	1.70	34,022	23.50
Spanish (es)	5,022,610	3,722,927	6,490,447	1.74	–	–
Dutch (nl)	4,416,028	3,817,696	6,456,175	1.69	27,720	26.25
Italian (it)	4,087,765	3,541,031	5,423,837	1.53	27,172	29.00
Romanian (ro)	3,009,318	2,697,720	3,384,256	1.25	24,375	27.25

Table 1: Statistics of BabelNet for the languages used in the experiments and coverage of the corpus with Babel synsets.

the lexico-semantic relations from WordNet and Wikipedia, but this information is not currently used in our approach, which focuses on the cross-language nature of synsets.

The left-hand side of Table 1 shows the key BabelNet figures for the seven languages used in our work. We observe a considerable gap between the number of lemmas covered in English and the remaining languages. However, as we show in Section 4.1., the difference does not translate into a significantly different coverage of the corpus. In what follows, languages are named by the ISO 329-1 code shown in the same table.

3. Seq2seq Neural Machine Translation

State-of-the-art NMT systems are seq2seq architectures with recurrent neural networks (RNN) (Cho et al., 2014; Sutskever et al., 2014; Bahdanau et al., 2014). Briefly, an encoder projects source sentences into an embedding space and a decoder generates target sentences from the encoder embeddings.

Let $s = (x_1, \ldots, x_n)$ be a source sentence of length n. The encoder encodes s as a context vector at each word position, $\{\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_n\}$, where each component is obtained by concatenating the forward ($\overrightarrow{\mathbf{h}}_i$) and backward ($\overleftarrow{\mathbf{h}}_i$) encoder RNN hidden states:

$$\mathbf{h}_i = \left[\overleftarrow{\mathbf{h}}_i, \overrightarrow{\mathbf{h}}_i\right] \tag{1}$$

with recurrent units

$$\overleftarrow{\mathbf{h}}_i = \tanh\left(\mathbf{W_x} \,\|_{k=1}^{|F|}\, E_{xk}x_{ik} + \mathbf{U_x}\overleftarrow{\mathbf{h}}_{i-1}\right) \tag{2}$$

$$\overrightarrow{\mathbf{h}}_i = \tanh\left(\mathbf{W_x} \,\|_{k=1}^{|F|}\, E_{xk}x_{ik} + \mathbf{U_x}\overrightarrow{\mathbf{h}}_{i-1}\right), \tag{3}$$

where $\mathbf{W}_x$ and $\mathbf{U}_x$ are trainable weight matrices, $\mathbf{E_x}$ is the matrix of the source embeddings, and $\|$ is the concatenation operator. In the most simple case, the system is only trained with words so, $|F| = 1$, and $\mathbf{E_x}$ corresponds to the matrix of word embeddings. Semantic information can be included as additional factors to the word representations. In this case, one considers two factors, $|F| = 2$, and concatenates synset embeddings to word embeddings which are learned independently. Other features and kinds of operations such as sum or multiplication could be used, the ones described here are those applied in our experiments.

Defined in this way, factors do not affect the decoding architecture. Let $t = (y_1, \ldots, y_m)$ be a target sentence of length m. The recurrent hidden state of the decoder $\mathbf{z}_j$ is computed using its previous hidden state $\mathbf{z}_{j-1}$, as well as the continuous representation of the previous target word $\mathbf{t}_{j-1}$ and the weighted context vector $\mathbf{q}_j$ at time step j:

$$\mathbf{z}_j = g(\mathbf{z}_{j-1}, \mathbf{t}_{j-1}, \mathbf{q}_j) \tag{4}$$

$$\mathbf{t}_{j-1} = \mathbf{E_y} \cdot \mathbf{y}_{j-1}, \tag{5}$$

where g is a non-linear function and $\mathbf{E_y}$ is the matrix of the target embeddings. The weighted context vector $\mathbf{q}_j$ is calculated by the *attention mechanism* as described in Bahdanau et al. (2014). Its function is to assign weights to the context vectors in order to selectively focus on different source words at different time steps of the translation and it is calculated as follows:

$$a(\mathbf{z}_{j-1}, \mathbf{h}_i) = \mathbf{v}_a \cdot \tanh(\mathbf{W}_a \cdot \mathbf{z}_{j-1} + \mathbf{U}_a \cdot \mathbf{h}_i) \tag{6}$$

$$\alpha_{ij} = \mathrm{softmax}\left(\mathrm{a}(\mathbf{z}_{\mathrm{j}-1}, \mathbf{h}_{\mathrm{i}})\right), \quad \mathbf{q}_j = \sum_i \alpha_{ij}\mathbf{h}_i \tag{7}$$

Finally, the probability of a target word is given by the following softmax activation (Sennrich et al., 2017):

$$p(y_j|\mathbf{y}_{<j}, \mathbf{x}) = p(y_j|\mathbf{z}_j, \mathbf{t}_{j-1}, \mathbf{q}_j) = \mathrm{softmax}\left(\mathbf{p}_j\mathbf{W}\right) \tag{8}$$

$$\mathbf{p}_j = \tanh\left(\mathbf{z}_j\mathbf{W}_{p1} + \mathbf{E_y}[y_{j-1}]\mathbf{W}_{p2} + \mathbf{q}_j\mathbf{W}_{p3}\right) \tag{9}$$

where $\mathbf{W}_{p1}, \mathbf{W}_{p2}, \mathbf{W}_{p3}, \mathbf{W}$ are trainable matrices.

The number of target words in these systems is limited by the complexity of the training. The larger the vocabulary is, the higher the computing time and the memory needed. Usually, less than 100 k unique words are used.

4. Experimental Settings

4.1. Corpora

We use the *en-de-ro-it-nl* TED corpus provided for the IWSLT 2017 multilingual task (Cettolo et al., 2017). It includes 9161 talks in five languages, 4,380,258 parallel sentences when all the language pairs are considered. The intersection of talks among languages is high, 7945 documents are common to all of them, and therefore the same sentence is available in multiple languages. Notice that the size of the corpus is small as compared to standard collections of bilingual corpora —the WMT[1] en-fr set contains 36 M sentence pairs and the *en-de* one 5 M for instance. However, its multilingual nature makes it adequate for this study.

[1] `http://statmt.org/wmt14/translation-task.html`

SYSTEM w: $< 2en >$ es war ein riesiger Erfolg
$< 2en >$ è stato un enorme successo

SYSTEM wb: $< 2en >$|- es|- war|- ein|- riesiger|- Erfolg|bn:15350982n
$< 2en >$|- è|bn:00083181v stato|bn:00083181 un|- enorme|bn:00102268a successo|bn:00078365n

SYSTEM b: PRONOUN VERB DETERMINER ADJECTIVE bn:15350982n
bn:00083181v bn:00083181v DETERMINER bn:00102268a bn:00078365n

Figure 1: Example sentence of *tst2010* in German and Italian encoded to be translated into English for the three systems introduced in Section 4.2.: w, wb and b.

$< 2en >$|- es|- **war|bn:00083181** ein|- riesiger|- **Erfolg|bn:15350982n**
$< 2de >$|- and|- it|- **was|bn:00083181v** a|- huge|bn:00098905a **success|bn:00075023n**
$< 2en >$|- ed|- **è|bn:00083181v stato|bn:00083181v** un|- enorme|bn:00102268a **successo|bn:00078365n**
$< 2en >$|- en|- het|- **was|bn:00083181v** een|- groot|- **succes|bn:06512571n**
$< 2en >$|- și|bn:00012706n a|- **fost|bn:00083181v** un|- mare|bn:00098342a **succes|bn:00075024n**

Figure 2: Sentence extracted from *tst2010* in the five languages of the TED corpus $en\text{-}de\text{-}ro\text{-}it\text{-}nl$. The encoding as input to system wb shows differences and similarities of Babel synsets among languages.

We annotate the documents with a coarse-grained part of speech tagset (PoS), lemma and Babel synsets. Our PoS tag set consists of 10 elements defined to be compatible with the one in the BabelNet ontology {NOUN, VERB, PREPOSITION, PRONOUN, DETERMINER, ADVERB, ADJECTIVE, CONJUNCTION, ARTICLE, INTERJECTION}. The IXA pipeline (Agerri et al., 2014) is used to annotate en, de, es and fr documents with PoS and TreeTagger (Schmid, 1994) for nl, ro and it. The original tags are then mapped to our common reduced tagset[2]. The same tools are used to annotate the texts with lemmas.

Only a subset of PoS tags is enriched with their synset information. We select (*i*) nouns —including named entities, foreign words and numerals, (*ii*) adjectives, (*iii*) adverbs and (*iv*) verbs. In addition, we explicitly mark negation particles with a tag NEG and include them here to account for their semantics. Since a word can have several Babel synsets, we retrieve a synset according to the lemma and PoS of a word. In case there is still ambiguity, we select the BabelNet ID as the ID according to the BabelNet ordering of IDs: "*(a) puts WordNet synsets first; (b) sorts WordNet synsets based on the sense number of a specific input word; (c) sorts Wikipedia synsets lexicographically based on their main sense*" (Navigli, 2013, p. 35).

With this procedure, 26.5% of the corpus is covered by synset identifiers and the remaining 73.5% only by PoS tags, where the coverage per language is similar and ranges from 23.5% to 29.0%, see Table 1.

4.2. NMT Systems

Our systems are NMT engines trained with Nematus (Sennrich et al., 2017). We train three systems:

w**:** A many-to-many NMT engine trained on parallel corpora for the several language pairs simultaneously. As in Johnson et al. (2017) and similarly to Ha et al. (2016), the engine is trained with the only addition of a tag in the source sentence to account for the target language "<2trg>". We only consider those sentences with less than 50 tokens for training, that is 2,113,917 parallel sentences (39,393,037 tokens).

wb**:** A many-to-many factored NMT engine (Sennrich and Haddow, 2016) trained on the same corpus as before but enriched with Babel synsets as an additional factor.

b**:** A one-to-one NMT system trained on the part of the corpus with English as target. All the source languages are encoded as Babel synsets instead of words; for any word without a known synset, we use the PoS. This way, we obtain 868,226 parallel sentences (15,684,750 tokens).

Figure 1 shows example sentences coded according to each system.

Regarding the system's parameters, we use a learning rate of 0.0001, Adadelta optimisation, 800 hidden units, a mini-batch size of 100, and drop-out only for hidden layers and input embeddings. We also tie the embeddings in the decoder side to reduce the size of the translation models. The dimension of the embeddings is always 506; for the factored system wb we reserve 300 dimensions for words and 206 for synsets. All the systems have a maximum common vocabulary of 150 k. Systems w and wb add 2 k for subword units segmented using Byte Pair Encoding (BPE) (Sennrich et al., 2016). Subwords in the source sentence are annotated with the same factors as the corresponding complete word. There is no BPE segmentation in system b. For decoding, we use an ensemble with the last four models at intervals of 10000 mini-batches and a beam size of 10.

5. Results and Discussion

Table 2 shows the translation performance of the three systems defined in the previous section on the 2010 IWSLT test set (*tst2010*), a test set build up with unseen TED talks. Systems are trained on en, de, ro, it and nl data (top rows); fr and es (bottom rows) have not been seen in training and correspond to what we call beyond-zero-shot translation. Boldfaced scores in the table mark the best system for a

[2] The mappings and the full annotation pipeline can be obtained here: https://github.com/cristinae/BabelWE

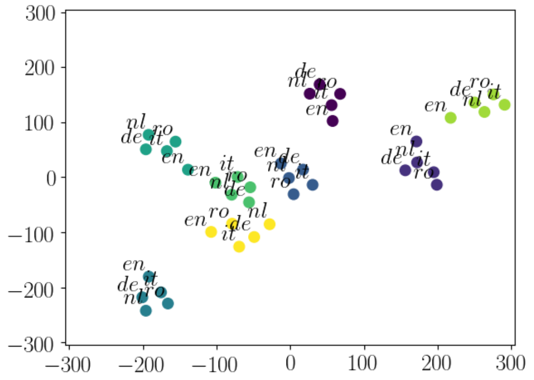

Figure 3: 2D t-SNE representation of the context vectors of the first 8 source sentences of *tst2010* for system wb. The same sentence has the same colour in different languages.

language pair and, when systems are not statistically significantly different from the best one with at least a p-value of 0.01, we mark them as well. Bootstrap resampling is used to estimate statistical significance (Koehn, 2004).

For the languages with training data, we observe that the addition of cross-language synsets as factors moderately improves the translation quality as measured by BLEU (Papineni et al., 2002) and METEOR (Denkowski and Lavie, 2014). Both are lexical metrics that essentially count the number of n-gram matches between a system translation and the reference translation(s). We use the version of METEOR that considers matches between words, stems, synonyms and paraphrases.

Cross-language IDs are of special interest to ML-NMT systems because they can help to cluster together sentences according to their meaning and irrespective of their language. Since a word vector in our factored NMT (wb) has the top elements representing the word itself and the bottom ones representing the synset ID, different words with same synset share a common part of the representation. In fact, such a clustering is already done in ML-NMT systems, but the quality is worse the more distant languages are (España-Bonet et al., 2017) and synsets can help to overcome this distance. In our case, by using the ML-TED corpus, we train the systems with the same sentence in different languages, so this grouping is already eased by construction, as becomes apparent through a graphical representation of the sentences. Figure 3 depicts a 2D t-SNE representation (Van Der Maaten, 2014) of the context vectors of 8 sentences of the test set with the wb system. The clustering by sentence (colour) is evident in the plot but we obtain very similar clustering visualisations with the w system. Since the initial grouping is already good, the addition of the synsets improves the translation by only 0.23 points on average.

The quality of the synset annotation is also relevant for performance. One of the major issues in our setting is the fact that the top synset in a language does not always correspond to the top synset in another one. The example sentence in Figure 2 is an extreme case where the word *success* has five different IDs depending on the language. The verb *to be* on the contrary, is identified as *bn:00083181v* in all of them.

	BLEU			METEOR		
	w	wb	b	w	wb	b
$de2en$	**32.6**	**33.0**	17.5	33.1	**33.5**	24.2
$it2en$	**33.5**	**33.2**	21.4	**33.9**	**34.0**	27.4
$nl2en$	36.2	**36.6**	15.0	34.7	**34.9**	21.5
$ro2en$	**34.3**	**34.8**	19.6	**34.4**	**34.6**	25.9
$fr2en$	2.4	5.1	**7.3**	11.2	16.7	**17.5**
$es2en$	3.1	6.7	**11.3**	12.0	18.4	**20.7**

Table 2: Automatic evaluation of the systems defined in Section 4.2. on *tst2010*. Boldfaced scores indicate the best systems; systems not statistically significantly different from the best one ($p = 0.01$) are also boldfaced.

Improving the cross-linguality in the synset annotation is a key aspect to achieve further improvements. Besides, as stated in Section 4.1., we did not perform any word sense disambiguation for retrieving the synset but took the top ID, so we are missing relevant information for translation which could also help to gather the truly interlingua IDs.

Even with these identified limitations, the factored system wb already improves on the word system w and this is even more evident in the case of languages that have not been seen at training time. The last two rows in Table 2 display the results when translating from unseen es and fr into en. In this case, the system does not have the vocabulary of the language, so a BLEU score of 2.4 ($fr2en$) and 3.1 ($es2en$) is obtained mainly thanks to identical named entities, digits and cognates between the languages. The inclusion of synsets is in this case more important, because words sharing the synset ID can be now translated and that increases the BLEU scores to 5.1 ($fr2en$) and 6.7 ($es2en$), +3.2 BLEU points in average. Similar differences are seen with METEOR. Still the numbers are far from those obtained for languages seen in training.

System b is totally different. Here the source words are not used at all and we keep *what is left after removing the structural particularities of the base language* as Richens (1956) suggested to encode a source sentence. For a language pair with parallel corpora this representation is clearly worse than the original one because all the morphological information and even the semantics of prepositions, determiners and conjunctions is lost. However, the semantics of content words is kept in an interlingual way and that improves the translation of unseen languages, +6.5 BLEU and +7.5 METEOR points on average as shown in Table 2.

Comparing b with similar systems trained on monolingual data, we observe that the translation is possible because we use multiple languages on the source side, and the network learns different combinations to encode the same expression. For instance, both *“PRONOUN VERB DETERMINER ADJECTIVE bn:15350982n”* and *”bn:00083181v bn:00083181v DETERMINER bn:00102268a bn:00078365n“* should be translated as *”it was a huge success“* (Figure 1). This diversity is important to accommodate new languages.

Strengths and weaknesses of the three systems can be seen in the example translation shown in Figure 4. For the languages with training data, w and wb provide the same

	SYSTEM w:	SYSTEM wb:	SYSTEM b:
$de2en$	and it was a huge success	and it was a huge success	and it 's a huge success
$it2en$	and it was a huge success	and it was a huge success	and it was a huge success
$nl2en$	and it was a big success	and it was a big success	and this is a huge success
$ro2en$	and it was a great success	and it was a great success	and it was a great success
$fr2en$	it 's the facade of a great success	and the Khan has been a great winner	but there was a big winner
$es2en$	y is a great deal	y is a great marker	but it 's a great winner

Figure 4: Example sentence of *tst2010* in the languages of the study translated into English by the three systems introduced in Section 4.2.

translation for this simple sentence. For fr and es, where there is no training data, some of the words are cognates and have been seen in other languages (*gran/*es, *grand/*fr, *succes/*fr; *gran/it*, *grand/en*, *succes/ro*) while some others have not (*fue/es*, *ça/*fr, *été/*fr). In the latter case the w system just builds the translation as the concatenation of seen BPE subunits (*ça a été/*fr ⇒ *it's the facade of/en*), while the wb system is able to recognise the verb thanks to the synset (*été|bn:00083181v* ⇒ *has been*). As before, b behaves differently. When the synset is correctly assigned, the system can translate the adjective (*huge*, *big*, *great*) even if the ID differs for each source language. As shown in Figure 1, *riesiger* in the German sentence could not be mapped to a synset, so system b translates it from the source token *ADJECTIVE*. In this particular case the translation is correct because during training the system has learnt that *huge* is the most probable translation for *ADJECTIVE* when it goes before *Erfolg*. However, part-of-speech tags cannot always be translated properly and we obtain different choices for *CONJUNCTION* (*and*, *but*) and *PRONOUN* (*it*, *this*, *there*) depending on the sentence. Conjugations might not be correctly translated either: *VERB* (*'s*, *is*, *was*).

The previous example shows how in order to make the most of this architecture, one would need an additional abstraction step for non-content words and making morphology explicit in the source side, and then in the corresponding generation step in the decoder side. That would even allow to train a synset2target NMT system using only monolingual data. These refinements are left as future work.

6. Summary and Conclusions

We have shown two different ways to include the knowledge encoded in semantic networks in NMT systems. The first one, system wb, adds interlingual Babel synsets as a factor. This way, we obtain moderate improvements in ML-NMT translation for known languages, and more than 3 BLEU points for languages not seen in training. The second one, b, encodes the input as a sequence of Babel synsets completed with PoS tags and entirely ignoring the specific words of the source language. This way, we further improve translation for languages not seen in training (beyond zero shot) by more than 6.5 BLEU and 7.5 METEOR points on average.

The next natural step is to design these systems so that they can be trained on monolingual corpora only. To do this, we need first to better choose (i.e. properly disambiguate) the synset of a word so that it is the same irrespective of the language. Second, one needs to add abstraction and generation layers to deal with morphology and non-content words in the target language.

Notice that the methodology used benefits from the ability of seq2seq models to learn in multilingual settings, so it is not exclusive to NMT and it can be also applied to multilingual/crosslingual neural text summarisation or question answering systems for example.

7. Acknowledgements

The project on which this paper is based was partially funded by the German Federal Ministry of Education and Research under the funding code 01IW17001 (Deeplee) and by the Leibniz Gemeinschaft via the SAW-2016-ZPID-2 project (CLuBS). Responsibility for the content of this publication is with the authors. The authors are grateful to R. Navigli for the useful discussions and comments.

8. Bibliographical References

Agerri, R., Bermudez, J., and Rigau, G. (2014). IXA pipeline: Efficient and Ready to Use Multilingual NLP Tools. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland, may. European Language Resources Association (ELRA).

Artetxe, M., Labaka, G., Agirre, E., and Cho, K. (2017). Unsupervised neural machine translation. *CoRR*, abs/1710.11041.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural Machine Translation by Jointly Learning to Align and Translate. *CoRR*, abs/1409.0473, September.

Bojar, O., Chatterjee, R., Federmann, C., Graham, Y., Haddow, B., Huck, M., Koehn, P., Logacheva, V., Monz, C., Negri, M., Post, M., Rubino, R., Specia, L., and Turchi, M. (2017). Findings of the 2017 Conference on Machine Translation. In *Proceedings of the Second Conference on Machine Translations (WMT 2017)*, pages 169–214, September.

Cettolo, M., Federico, M., Bentivogli, L., Niehues, J., Stüker, S., Sudoh, K., Yoshino, K., and Federmann, C. (2017). Overview of the IWSLT 2017 Evaluation Campaign. In *Proceedings of the 14th International Workshop on Spoken Language Translation (IWSLT)*, Tokyo, Japan, December.

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–

1734, Doha, Qatar, October. Association for Computational Linguistics.

Denkowski, M. and Lavie, A. (2014). Meteor Universal: Language Specific Translation Evaluation for Any Target Language. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 376–380, Baltimore, Maryland, USA, June. Association for Computational Linguistics.

Du, J., Way, A., and Zydron, A. (2016). Using babelnet to improve oov coverage in smt. 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).

España-Bonet, C., Varga, A. C., Barrón-Cedeño, A., and van Genabith, J. (2017). An Empirical Analysis of NMT-Derived Interlingual Embeddings and their Use in Parallel Sentence Identification. *IEEE Journal of Selected Topics in Signal Processing*, 11(8):1340–1350, December.

Ha, T., Niehues, J., and Waibel, A. H. (2016). Toward Multilingual Neural Machine Translation with Universal Encoder and Decoder. In *Proceedings of the International Workshop on Spoken Language Translation*, Seattle, WA, November.

Johnson, M., Schuster, M., Le, Q. V., Krikun, M., Wu, Y., Chen, Z., and1 Fernanda B. Viégas, N. T., Wattenberg, M., Corrado, G., Hughes, M., and Dean, J. (2017). Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation. *Transactions of the Association for Computational Linguist*, 5:339–351, October.

Koehn, P. (2004). Statistical Significance Tests for Machine Translation Evaluation. In Dekang Lin et al., editors, *Proceedings of EMNLP 2004*, pages 388–395, Barcelona, Spain, July. Association for Computational Linguistics.

Lample, G., Denoyer, L., and Ranzato, M. (2017). Unsupervised machine translation using monolingual corpora only. *CoRR*, abs/1711.00043.

Navigli, R. and Ponzetto, S. P. (2012). BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250.

Navigli, R. (2013). A quick tour of babelnet 1.1. In *CICLing (1)*, volume 7816 of *Lecture Notes in Computer Science*, pages 25–37. Springer.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the Association of Computational Linguistics*, pages 311–318.

Richens, R. H. (1956). Preprogramming for mechanical translation. *Mechanical Translation*, 3(1):20–25, July.

Schmid, H. (1994). Probabilistic part-of-speech tagging using decision trees. In *International Conference on New Methods in Language Processing*, pages 44–49, Manchester, UK.

Sennrich, R. and Haddow, B. (2016). Linguistic input features improve neural machine translation. In *Proceedings of the First Conference on Machine Translation*, pages 83–91, Berlin, Germany, August. Association for Computational Linguistics.

Sennrich, R., Haddow, B., and Birch, A. (2016). Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL, August 7-12, 2016, Berlin, Germany, Volume 1*.

Sennrich, R., Firat, O., Cho, K., Birch, A., Haddow, B., Hitschler, J., Junczys-Dowmunt, M., Läubli, S., Miceli Barone, A. V., Mokry, J., and Nadejde, M. (2017). Nematus: a toolkit for neural machine translation. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 65–68, Valencia, Spain, April. Association for Computational Linguistics.

Srivastava, A., Rehm, G., and Sasaki, F. (2017). Improving machine translation through linked data. *The Prague Bulletin of Mathematical Linguistics (PBML)*, 108:355–366, 6.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc.

Van Der Maaten, L. (2014). Accelerating t-SNE Using Tree-based Algorithms. *Journal of Machine Learning Research*, 15(1):3221–3245, January.

Bianet: A Parallel News Corpus in Turkish, Kurdish and English

Duygu Ataman
Fondazione Bruno Kessler
Università degli Studi di Trento
Via Sommarive 18, Trento, Italy
ataman@fbk.eu

Abstract

We present a new open-source parallel corpus consisting of news articles collected from the Bianet magazine, an online newspaper that publishes Turkish news, often along with their translations in English and Kurdish. In this paper, we describe the collection process of the corpus and its statistical properties. We validate the benefit of using the Bianet corpus by evaluating bilingual and multilingual neural machine translation models in English-Turkish and English-Kurdish directions.

Keywords: Parallel corpora, Machine Translation, Turkish, Kurdish, Low-resource Languages

1. Introduction

Machine translation (MT) is the task of translating a sequence of text to a given language. Current approaches to MT are based on statistical learning, where a probabilistic model learns to generate outputs based on previous observations of translation examples in the given language direction, often referred as *parallel corpora*. By learning a statistical translation model using these corpora and using it to predict translations to future words, MT allows to automate the translation task between any pair of languages. The main advantage of statistical MT is the ability to obtain translations without requiring prior knowledge on the specific language or using any linguistic tools. On the other hand, in order to build robust and reliable translation systems it is required to use a sufficient amount of parallel corpora in the given domain of the translation that should provide observations of words and their translations in various but terminologically similar contexts.

However, a crucial problem in statistical MT is the lack of availability of parallel corpora in many translation domains and directions. One important translation domain is news, as it has many applications in the industry, although the available parallel data in this domain is very limited in many languages. Turkish, for instance, is a language spoken by around 67 million people in the Republic of Turkey, although it suffers from the lack of publicly available data resources. The only parallel news corpus in the English-Turkish language direction is the South-east European Times (SETIMES) corpus (Tyers and Alperen, 2010)). Another language spoken in Turkey with a lack of any parallel corpora is Kurdish (Northern), with around 8 million speakers (Simons and Fennig, 2017). Two languages are quite different by nature, Kurdish is an Indo-European language whereas Turkish belongs to the Turkic language family. However, they share many common words in their vocabularies due to a long history of social interaction between the speakers of the two languages.

In this paper, we address this problem and present a new parallel corpus consisting of sentence-aligned news articles in Turkish, Kurdish (in the Latin script) and English. Our corpus consists of collected articles from the Bianet[1] online newspaper, a website that publishes daily news on politics, law, economy and cultural events in Turkey. All articles are originally written in Turkish, while some of them are also translated into English or Kurdish (or both) by human translators, thus, they are usually available in multiple languages. We construct our corpus using online crawling tools and further process the collected sentences to check if they are correctly aligned. The retrieval process is described in Section 2., whereas the statistical properties of the resulting corpora are presented in Section 3.. The major part of the corpus is the English-Turkish portion, which provides additional data to translation tasks in the news domain for this language pair. Therefore, we illustrate the benefit of using the Bianet corpus in the English-Turkish language direction by building a news domain neural MT system (Bahdanau et al., 2014) and evaluate it on the data sets from an official MT evaluation campaign. Although the portions that contain Kurdish translations are not sufficient enough to build a stand-alone MT system, given the low-resource feature of Kurdish, these corpora can still be useful in applications in MT in English-Kurdish and Turkish-Kurdish language pairs, such as for building multi-lingual translation models or fine tuning generic domain models. Using this approach, we make use of the Bianet corpus for building multi-lingual neural MT models and evaluate them in English-Turkish and English-Kurdish translation directions. The findings of our experiments, given in Section 4., show that using our corpus for training MT systems in the news domain can aid in obtaining better translations, whereas both languages can be improved by means of multi-lingual models. The Bianet corpus is available online and can be used for any research purpose[2].

2. Collecting the Corpus

The collection of the Bianet corpus consists of mainly three steps:

[1] Available at the website www.bianet.org

[2] The Bianet corpus can be downloaded from **https://d-ataman.github.io/bianet**

Language	Number of Sentences	Number of Tokens	Vocabulary Size
Turkish-English	35,080	741,080 (*EN*) - 582,783 (*TR*)	61,517 (*EN*) - 103,812 (*TR*)
English-Kurdish	6,486	139,334 (*EN*) - 126,350 (*KU*)	19,362 (*EN*) - 21,462 (*KU*)
Turkish-Kurdish	7,390	121,119 (*TR*) - 142,668 (*KU*)	32,064 (*TR*) - 23,333 (*KU*)

Table 1: Statistical Properties of the Corpus. *EN:* English side. *TR:* Turkish side. *KU:* Kurdish side.

- Crawling the Turkish news articles in the newspaper domain
- Retrieving the document-level translations and building comparable documents in each language pair
- Alignment of each sentence in the document-aligned corpora

In this section, we present the details of the implementation of all the steps that resulted in the Bianet corpus.

2.1. Crawling the news articles

The web crawling is implemented using Scrapy[3], an open-source library implemented in Python for extracting data from web pages. The crawling is accomplished using *Spiders*, custom classes that allow to define ways to crawl pages, such as by following links or extracting portions with corresponding HTML tags. In order to crawl the news site of Bianet, we build a news Spider that continuously reads each article in Turkish by iterating over pages. From each extracted article link, the crawling continues if and only if the next article link is within the website domain and is relevant to the list of categories. The allowed categories are politics, culture, law, human rights, women, environment, society, art, sports and culture.

For each retrieved article, the spider processes the web page source to detect if there are any links to available translations of the article. Most articles contain a link to the English and Kurdish versions in the beginning of the page, with an HTML tag that is easily discovered, as in *'Click for English'*. After each article is crawled, the web pages that represent its translations are also crawled to form a group of two or three articles, each in a different language. The program extracts the raw text body from each article in the group and then saves them using the same article id. This operation is repeated for all articles in the website until all articles that fit the relevant categories are crawled. The overall process of crawling the website domain results in 3,214 Turkish articles which have English translations, 824 Turkish articles with Kurdish translations, and 845 English articles which also have Kurdish translations.

2.2. Building Comparable Documents

The articles crawled as described in Section 2.1. are later processed and combined into a collection of three portions representing each language pair. This process is quite straight-forward as our implementation of the crawling step simultaneously crawls and saves each translation in the same id as the original article. The files are cleaned and empty lines are removed before we proceed to build sentence-aligned corpora.

2.3. Sentence Alignment

The comparable corpora obtained by crawling the web domain and cleaning the files are later transformed into parallel corpora using a sentence aligner. In our study, we use the HunAlign sentence aligner[4] (Varga et al., 2007). Hunalign is a tool for building bilingual text at the sentence level. The program takes as input two comparable documents in different languages and then generates bilingual sentence pairs.

3. Statistical Properties

In this section, we present the Bianet corpus which consists of 35,080 sentences, and around 1,3 million tokens. The Turkish side of the parallel corpus contains a total vocabulary of 103,812 unique words, which is a significant vocabulary contribution for a sparse language like Turkish. The English-Kurdish and Turkish-Kurdish portions are rather smaller compared to the first portion. The English-Kurdish corpus contains 6,486 sentences, whereas the Turkish-Kurdish portion contains 7,390. Further information of the statistical properties of the corpus can be found in Table 1.

4. Experiments

In order to illustrate the contribution of the Bianet corpus, we conduct statistical MT experiments in the English-Turkish and English-Kurdish language directions. We evaluate the benefit of using our corpus by including it in the training of models based on neural MT, the state-of-the-art method in statistical MT (Bahdanau et al., 2014). We first evaluate the quality of the translations in a English-Turkish translation model where we show the improvement on the output accuracy with the addition of the Bianet corpus on the translation model trained on the news domain. In the second stage, since both languages are low-resource, we train multi-lingual neural MT systems based on the approach of (Lakew et al., 2017), in order to further improve the translation quality. This section presents the details of these experiments.

4.1. Data

In English-Turkish experiments using bilingual neural MT, we build two bilingual neural MT systems in the news domain, one system only using the SETIMES corpus (Tyers and Alperen, 2010), and a second system using both SETIMES and Bianet corpora. We evaluate the two models using the official news development and testing sets from

[3] An open source data extraction framework, available at https://scrapy.org

[4] Available at http://mokk.bme.hu/resources/hunalign

Data set	Corpus	Language	Sentences	Words
Parallel Data (Translation Model)	SETIMES	English-Turkish	205,706	5,107,219 (*EN*) - 4,589,614 (*TR*)
	Bianet	English-Turkish	35,080	741,080 (*EN*) - 582,783 (*TR*)
	Bianet	English-Kurdish	6,486	139,334 (*EN*) - 126,350 (*KU*)
	Bianet	Kurdish-Turkish	7,390	142,668 (*KU*) - 121,119 (*TR*)
Dev	WMT dev2016	English-Turkish	1,001	22,136 (*EN*) - 16,954 (*TR*)
Test	WMT test2016	English-Turkish	3,000	66,394 (*EN*) - 54,128 (*TR*)

Table 2: Data sets used in the English-Turkish Experiments. *EN:* English side. *TR:* Turkish side.

Data set	Corpus	Language	Sentences	Words
Parallel Data (Translation Model)	Ubuntu & GNOME	English-Kurdish	65,357	206,855 (*EN*) - 219,279 (*KU*)
	Bianet	English-Kurdish	6,486	139,334 (*EN*) - 126,350 (*KU*)
	SETIMES & Bianet	English-Turkish	240,786	5,848,299 (*EN*) - 5,172,397 (*TR*)
	Bianet	Turkish-Kurdish	7,390	142,668 (*KU*) - 121,119 (*TR*)
Dev	Sampled from Bianet	English-Kurdish	500	11,311 (*EN*) - 5,399 (*KU*)
Test	Sampled from Bianet	English-Kurdish	500	11,174 (*EN*) - 5,696 (*KU*)

Table 3: Data sets used in the English-Kurdish Experiments. *EN:* English side. *KU:* Kurdish side.

WMT 2016[5] (Bojar et al., 2016). In the multilingual neural MT systems in the English-Turkish direction, we also use the English-Kurdish and Kurdish-Turkish portions of the Bianet corpus. Similarly, in English-Kurdish experiments, we build a generic neural MT model using a training set consisting of the only publicly available English-Kurdish parallel datasets, Ubuntu and GNOME (Tiedemann, 2012). Since there are no available official evaluation data sets, we sample the development and the testing sets from the Bianet corpus so that they reflect the news domain. We then build a multilingual neural MT model using the English-Kurdish, English-Turkish and Turkish-Kurdish portions of the Bianet corpus. The details of the data used in the experiments can be seen in Tables 2 and 3.

4.2. Models

The neural MT models used in the evaluation are based on the Nematus toolkit (Sennrich et al., 2017). They have a hidden layer and embedding dimension of 1024, a mini-batch size of 100 and a learning rate of 0.01. The dictionary size is 40,000 for both the source and target languages. For vocabulary reduction we use the subword segmentation method described in (Ataman et al., 2017), with the default settings and a target vocabulary size of 40,000. We train the models using the Adagrad (Duchi et al., 2011) optimizer, and a dropout rate of 0.1 in the input and output layers and 0.2 in the embeddings and hidden layers. During training, we shuffle the data at each epoch for a total of 50 epochs and then choose the model with the highest performance on the development set for translating the test set. We use the BLEU (Papineni et al., 2002) and chrF3 (Popovic, 2015) automatic evaluation metrics and the Multeval multeval significance test for evaluating the accuracy of the models.

In English-Turkish translation, we train two models using two different parallel corpora, one only using the SETIMES corpus, called as the *Baseline Model News*, and a second model that is trained using also the Bianet corpus, referred to as *Combined Model News*. This allows us to illustrate the benefit of using our corpus for Turkish translation. The multi-lingual model is referred to as *Multilingual Model News*. In English-Kurdish translation, we build two models, one generic bilingual English-Kurdish model, called *Bilingual Model Generic*, which uses only the English-Kurdish parallel corpora. We also build one multilingual Turkish-Kurdish-English neural MT system, *Multilingual Model Generic*, which uses all available corpora. Since the already available English-Kurdish data in the IT domain (Ubuntu & GNOME) are not sufficient to build a reliable neural MT model in the news domain, the generic models can better illustrate the performance of MT systems in this language direction.

4.3. Results

The translation accuracy obtained on the WMT Turkish testing sets are given in Table 4. The model using the extended parallel training corpus achieves a significant improvement of **2.27 BLEU** and **0.0204 chrF3** points over the baseline model trained using only previously available SETIMES corpus. The significant improvement of 19.74% on the given evaluation task verifies the quality of human translations in the Bianet corpus and confirms the benefit of its usage for training MT systems in the news domain. Yet, in English-Turkish translation, the best performance is achieved with the multilingual model which incorporates all portions of the Bianet corpus, which outperforms the baseline model by **2.42 BLEU** and **0.0221 chrF3** points.

Similarly, in English-Kurdish translation, as given in Table 5, the corpus shows promising application by allowing to generate translations in the news domain with a quality of **5.41 BLEU** and **0.2257 chrF3** points using only a generic model, which is trained on small corpora from different domains. Adding also the multi-lingual data in Turkish-Kurdish and Turkish-English from the Bianet corpus more-

[5] The Official Shared Task of MT of News, The First Conference on MT, 2016

System	Output Score	
	BLEU	**chrF3**
Baseline Model News	11.50	0.4139
Combined Model News	13.77	0.4343
Multilingual Model News	**13.92**	**0.4360**

Table 4: English-Turkish Experiment Results. *Baseline Model News:* The model built only using SETIMES corpus. *Combined Model News:* The model built using SETIMES and the English-Turkish portion of Bianet corpus. *Multilingual Model News:* The model built using SETIMES and all portions of the Bianet corpus. Best scores are in bold font. All improvements over the baseline are statistically significant (p-value $<$ 0.05).

System	Output Score	
	BLEU	**chrF3**
Bilingual Model Generic	5.41	0.2257
Multilingual Model Generic	**8.51**	**0.2406**

Table 5: English-Kurdish Experiment Results. *Bilingual Model Generic:* The model built using corpora in English-Kurdish. *Multilingual Model Generic:* The model built using English-Kurdish corpora and all languages in the Bianet corpus. Best scores are in bold font.

over increases this quality by **4.10 BLEU** and **0.0149 chrF3** points.

5. Conclusion

We have presented a new parallel corpus in the news domain that aims at improving the MT of Turkish and Kurdish, two very low-resource languages. Our parallel corpus is a collection of news articles retrieved from the online news magazine Bianet. We have described the process of collecting and building the corpus as well as its statistical characteristics. We have also evaluated the quality of the translations in the corpus and the advantage of using it in MT by means of a set of experiments that compare bilingual and multi-lingual neural MT models using parallel corpora with and without including the Bianet corpus. The experiment findings show that the addition of the Bianet corpus yields a significant improvement on the overall translation quality, proving that it could be useful for building MT systems in the given language pairs. Our corpus is available online for public use.

6. Bibliographical References

Ataman, D., Negri, M., Turchi, M., and Federico, M. (2017). Linguistically Motivated Vocabulary Reduction for Neural Machine Translation from Turkish to English. *The Prague Bulletin of Mathematical Linguistics*, 108(1):331–342.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv preprint arXiv:1409.0473*.

Bojar, O., Chatterjee, R., Federmann, C., Graham, Y., Haddow, B., Huck, M., Yepes, A. J., Koehn, P., Logacheva, V., Monz, C., et al. (2016). Findings of the 2016 Conference on Machine Translation. In *Proceedings of the 1st Conference on Machine Translation (WMT)*, pages 131–198.

Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. In *Journal of Machine Learning Research*, volume 12, pages 2121–2159.

Lakew, S. M., Lotito, Q. F., Negri, M., Turchi, M., and Federico, M. (2017). Improving Zero-shot Translation of Low-resource Languages. *Proceedings of the 14th International Workshop on Spoken Language Translation (IWSLT)*, pages 113–119.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). BLEU: A method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL)*, pages 311–318.

Popovic, M. (2015). chrf: Character n-gram F-score for Automatic MT Evaluation. In *Proceedings of the 10th Workshop on Statistical Machine Translation (WMT)*, pages 392–395.

Sennrich, R., Firat, O., Cho, K., Birch, A., Haddow, B., Hitschler, J., Junczys-Dowmunt, M., Läubli, S., Barone, A. V. M., Mokry, J., et al. (2017). Nematus: a toolkit for Neural Machine Translation. In *Proceedings of the 15th Annual Meeting of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 65–68.

Simons, G. F. and Fennig, C. D. (2017). Ethnologue: Languages of the World. *SIL International*, 20.

Varga, D., Halácsy, P., Kornai, A., Nagy, V., Németh, L., and Trón, V. (2007). Parallel Corpora for Medium Density Languages. In *Amsterdam Studies in the Theory and History of Linguistic Science Series 4*, volume 292, page 247.

7. Language Resource References

Jörg Tiedemann. (2012). *Parallel Data, Tools and Interfaces in OPUS.* European Language Resources Association (ELRA).

Tyers, Francis M. and Alperen, Murat Serdar. (2010). *South-east European Times: A parallel corpus of Balkan Languages.*

Active Learning for Tibetan Named Entity Recognition based on CRF

Fei-Fei Liu, Zhi-Juan Wang*
Minzu University of China, National Language Resource Monitoring & Research Center of Minority Langues ; Minzu University of China National Language Resource Monitoring & Research Center of Minority Langues
Beijing China, Beijing China,
Liufeifei_muc@163. com, wangzj. muc@gmail. com

Abstract

Named entity recognition (NER) is a major subtask of information extraction. Previous research tent to use huge amount of labeled data to train a classifier. But it is expensive for low resource languages One of the dominant problems facing Tibetan named entity recognition is the lack of training data. Active learning is a supervised machine learning algorithm which can achieve greater accuracy with fewer training labels. Active learning has been successfully applied to a number of natural language processing tasks, such as, information extraction, named entity recognition, text categorization, part-of-speech tagging, parsing, and word sense disambiguation. In this paper, we apply active learning based on Conditional Random Field (CRF) for Tibetan named entity recognition to minimize labeling effort by selecting the most informative instances to label. This paper proposes two kinds of query strategies, including Confidence, and Named Entity features. We compare the query strategies with the random method, and show that considerable performance improvements in reduce the human effort.

Keywords: Active learning, Tibetan Named Entity Recognition, Query Strategy,CRF

1. Introduction

Named entity recognition (NER) is one of the most elementary and core problems in natural language processing (NLP). There are supervised learning (SL), semi-supervised learning (SSL), unsupervised learning(UL) for named entity recognition. At present, supervised machine-learning methods in the task are in the leading position, such as Hidden Markov Model (HMM), Conditional Random Field (CRF), Support Vector Machine (SVM). The obstacle of supervised machine-learning methods is the great requirement of the annotated training data which is essential for achieving good performance. Building a high quality annotated corpora by hand is time-consuming and expensive. Because of the lack of corpora and person who understand those languages, creating training corpora for resource-scarce languages is particularly expensive.

Nowadays, named entity recognition had achieved good results in various languages, such as English. State-of-the-art NER systems for English produce nearhuman performance. For example, the best system entering MUC[1]-7. scored 93.39% of F-measure while human annotators scored 97.60% and 96.95%. However, Tibetan NER started late. It has yielded a great number of positive results， but it is still a new study field in which there are series of problems, such as the conflicts between Tibetan names and ordinary words, the misinterpretation of translations, and the difficulties in identifying Tibetan NE boundaries. The biggest reason for these problems is the lack of high quality annotated corpora.

There is a way, active learning, to solve this problem. Active learning is a special case of semi-supervised machine learning in which a learning algorithm is able to interactively query the user (or some other information source) to obtain the desired outputs at new data points(Settles B, 2010 ; Rubens N et al., 2015). We can use it to select the most informative samples for training. In this way, we will undoubtedly enhance the performance under cutting annotating cost.

Many existing researches show that active learning can be effectively reduce the training quantity in NLP task(Olsson F, 2009), such as information extraction(Culotta A et al., 2006), text categorization (Dasgupta S and Hsu D , 2006), named entity recognition(Kim S et al., 2006 ; Tomanek K and Hahn U, 2009 ; Ekbal A et al., 2016 ; Saha S, 2012). However, the majority of this literature focuses on biomedical domain or the official languages of a certain country. Those research showed that active learning can select the useful data from a huge pool of unlabeled documents. We can use this method in Tibetan named entity recognition.

In this paper, we propose an alternative active learning strategy for the Tibetan NER task. Without large-scale labeled data, the proposed method greatly reduces the training time and annotating cost. Two methods are presented, the first method is based on the confidence, the second is mixed the tags information.

There are two kinds of query strategies to Tibetan named entity recognition. One is by the degree of confidence measure, we choose the lowest part of the degree as it hard to process. Another would calculate uncertainty for the tag. Last, we get the most likely annotation results, for each result, the confidence scores can be calculated. And the uncertainty is quantified by the difference of confidence.

The organization of the paper is as follows. Following the introduction in Section 1. Section 2 presents the background ,including Tibetan named entity recognition and active learing. Section 3 presents active learing for Tibetan NER based on CRF. Section 4 shows oue experiment and discussion. Finally, we summarize in section 5.

2. Back ground

2.1 Tibetan Tibetan Named Entity Recognition

2.1.1 Introduction to Tibetan

Tibetan(བོད་ཡིག) refers to the use of Tibetan language Tibetan. The glyph structure is a letter as the core, the rest of the letters are based on this before and after the additional and overlapping from top to bottom, combined into a complete word table structure. Writing habits from

[1] Message Understanding Conference

left to right. The font is divided into "head" and "headless" two categories.

Tibetan is a phonetic alphabet, with 30 consonants and 4 vowels. One Tibetan syllable can have 1 to 7 basic characters, if you consider Sanskrit, characters may be more. The seven basic characters have a base character and a vowel , the other characters were added to the base word, the up, down, front, back, and then back.

There are fewer types of punctuation in Tibetan . Tibetan various syllables separate with a small point, this point named the syllable node (་). In addition to the syllable node, the most common punctuation is a single vertical line (།), as a full stop, colon and other situations. And the paragraph ends with a double vertical line (།།).

2.1.2 Methods of Tibetan NER

The methods of Tibetan NER can be divided into rule-based methods and based on supervised machine learning methods.

Rule-based methods

In the early days, the study of Tibetan NER was based on a rule-based approach. Yu et al. used a rule-based model based on case-auxiliary word and lexicon, and also adapt boundary information list static from large corpus to improve recognition(Yu HZ et al.,2010). And experiments shows that recall rate and precision are respectively 90.13% and 94.02% in the newspaper corpus, 85.67% and 88.20% in the website text. Sun et al. used the internal features of names, contextual features and boundary features of names, and establishes the dictionary and feature base of Tibetan names(Sun Y et al.,2010). The results prove the algorithm is effective with 0.8391 F-score. Dou et al. used the Statistical Method of Mutual Information to, combining the rules of lattice auxiliary and the dictionary of person names, F value in the test can be up to 93.55%(Dou R et al.,2010).

Supervised machine learning methods

After 2014, supervised machine learning methods are increasingly applied to Tibetan NER. Jia et al. came up with Maximum entropy(ME) and conditional random field(CRF), and the F-score of the recognition of names can be 92.08%(Jia et al.,2014). Hua et al. proposed a syllable features with Perceptron training model to identify Tibetan name entity with detail analysis NE structure rule and word segmentation ambiguity(Hua et al.,2014). The F-score of NE identification is 86.03% for the test set. Kang et al. defined a feature tag set to fit in with the characters of Tibetan names, used CRF as tagging model to train and test corpus data(Kang et al.,2015). The highest F-score obtained in the experiment can reach 94.31%. Zhu et al. studied Tibetan name recognition technology using conditional random fields (CRF) principle， focuses on analysis of the internal structure of the Tibetan names, contextual features, feature selection and data preprocessing, etc. and evaluated the effectiveness of different features through experiments(Zhu et al.,2016). The recognition rate of Tibetan names can reach 80% of F-score.

2.1.3 Difficulties in Tibetan NER

Tibetan belongs to the Sino-Tibetan language family. In theory, the natural language processing methods used in Chinese can be used in Tibetan information processing. But in practice, it must be considered in the specific problems. The main difficulties in Tibetan NER are as follows:

Tibetan is a complex system of phonetic logic. The basic unit of the sentence is syllable. Syllables are separated by syllable node. One syllable or more syllables constitute words. There is no obvious mark between the word and next word. The boundaries of named entities are difficult to determine. And too few punctuation types, just single vertical line (།) and double vertical line (།།), will make the too long analysis object length, increasing the difficulty of recognition algorithm.

There is no morphological difference between named entities and unnamed entities in Tibetan. Unlike English, the person names, location names and organization names in English with the capitalized first letter, are easy to extract. And compared to Chinese person name, most of the Tibetans do not have the family name and the length of the name which can be from single syllable to twenty-six syllables.

The name dictionary, the labeled corpus and other related resources is insufficient. Nowadays, the main method of Tibetan Named Entity Recognition is supervised learning algorithms which require large-scale of labeled corpus. But Tibetan resource is not easy to obtain.

The biggest reason for these difficulties is the lack of Tibetan labeled corpus. We propose active learning to solve the problem.

2.2 Active Learning

In traditional supervised machine-learning, unlabeled data is selected for annotation at random under the huge amount of labeled data demand. Differently, the most useful data for the classifier are seriously selected in active learning.

2.2.1 Active Learning Examples

A learner may begin with a small number of instances in the labeled training set L, request labels for one or more carefully selected instances, learn from the query results, and then leverage its new knowledge to choose which instances to query next (Settles B, 2010). There is a Fig. 1 to indicate the typical pattern.

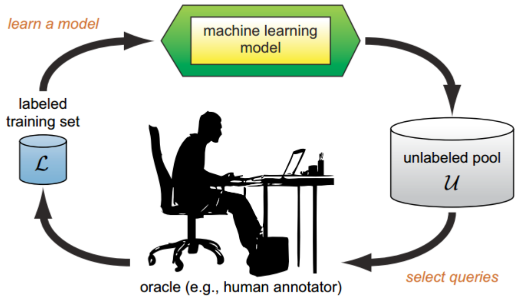

Figure 1: The typical pattern of active learning

And our frame of named entity recognition is by the following procedure. L is the labeled training set; U is the unlabeled data set; Q is the query strategy; C is the classifier for named entity recognition, in our work; N is the number of iterations.

Input：L, U, C, Q, N

Begin

```
  For i from 1 to N
    M=Train(C, L)
    /* Train classifier on L, get model M*/
    T=Test(C,M,U)
    /*with M， test U by C */
    T'=Select (Q, U|T)
    /*select useful by Q*/
    Label (T')
    /*query the human annotator for labeling*/
    L=L+ T';
    /*Add T' to L */
    U=U- T'; /* Delete T' from U*/
END
```

2.2.2 Query Strategy

There have been many proposed ways of formulating such query strategies.

Uncertainty Sampling

Perhaps the simplest and most commonly used query framework is uncertainty sampling (Lewis and Gale, 1994; Settles B, 2010). In this method, system queries the sentences about which it is least certain how to label the corpus, the criterion for the least confident strategy only considers information about the most probable label. It is straightforward and entropy is often used as an uncertainty measure.

Query-By-Committee

Query-by-committee (QBC) algorithm (Seung et al., 1992) as the more theoretically-motivated query selection framework is a good way to minimize the vision space. Each committee member is then allowed to vote on the labelings of query candidates. The most informative query is considered to be the instance about which they most disagree(Settles B, 2010).

Both of the above options are usual. There are other query strategies, for example, Expected Model Change, Expected Error Reduction, Variance Reduction, Density-Weighted Methods, etc.

In this paper, we propose two kinds of query strategies baded on the uncertainty sampling, including Confidence and Named Entity features. These query strategies are described in more detail in the subsequent sections.

3. Active Learning for Tibetan Named Entity Recognition based on CRF

3.1 Conditional Random Field

CRFs are a type of discriminative undirected probabilistic graphical model. It is used to encode known relationships between observations and construct consistent interpretations. It is often used for labeling or parsing of sequential data, such as natural language text. Specifically, CRFs find applications in shallow parsing, named entity recognition.

Lafferty, McCallum and Pereira define a CRF on observations X and random variables Y as follows:

Let G = (V , E) be a graph such that Y=(Y_v) v∈V, so that Y is indexed by the vertices of G. Then (X,Y) is a conditional random field when the random variables Y_v , conditioned on X, obey the Markov property with respect to the graph: p(Y_v |X, Y_w ,w≠v)=p(Y_v |X, Y_w ,w~v), where w~v means that w and v are neighbors in G.

What this means is that a CRF is an undirected graphical model whose nodes can be divided into exactly two disjoint sets X and Y, the observed and output variables, respectively; the conditional distribution p(Y|X) is then modeled.

By now, CRF has become a widely used technique which is applied in named entity recognition on low resource language, such as Hindi, Bengali, Tamil, and Telugu.

3.2 Tibetan Named Entity Recognition based on CRF

Tibetan NER can be defined as a sequence labeling problem for determining whether a observations belongs to a labeled set of markers. Suppose that a given marker sequence y= (y_1, y_2 , …, y_n) is labeled, n is the length of the sequence. The sequence of Tibetan NE is represented as w= (w_1, w_2,⋯, w_m), m is the length of the NE. The model of CRF is defined as follows:

$$p(y|w) = \frac{1}{Z(w)} \exp\left(\sum\nolimits_i \sum\nolimits_k \lambda\ _k f_k(y_i, y_{i-1}, w)\right)$$

Z(w) is normalization factor, determined by the observation sequence.

$$Z(w) = \sum\nolimits_y \exp\left(\sum\nolimits_k \lambda\ _k f_k(y_i, y_{i-1}, w)\right)$$

λ_k is the weight of the k-th function, f_k (y_i,$y_{(i-1)}$,w) is a characteristic function.

$$f_k(y_i, y_{i-1}, w) = \begin{cases} 1, \text{if } y_i = u \text{ and } y_{i-1} = v \\ 0, \text{otherwise} \end{cases}$$

3.3 Active Learning for Tibetan Named Entity Recognition

To solve the lack of Tibetan traing data,we present two kinds of query strategies in active learing .

3.3.1 Query Strategy based on Confidence

In **confidence**, we believe that the lower the confidence score of the sentence, the more difficult to identify for the classifier. This kind sentences need to manually participate in the annotation. And the confidence score can be calculated by Conditional probability. Give an input sequence x, in the situation that we have gotten the module, the P(y|x) is the Conditional probability that x corresponds to the tag sequence y. This probability can be regarded as confidence measure.

By using the equation for CRF (Lafferty et al. 2001) module, we can calculate the probability of any possible state sequence s given an input sequence. It is defined to be:

$$p(y|x) = \frac{1}{Z(x)} \prod_{t=1}^{T} \varphi_t\,(y_t, y_{t-1}, x_t)$$

$$\varphi_t(y_t, y_{t-1}, x_t) = \exp\left\{\sum_{k=1}^{K} \theta_k f_k(y_t, y_{t-1}, x_t)\right\}$$

To get the best sequence, we used the Viterbi algorithm, a dynamic programming algorithm for finding the most likely sequence of hidden states . So, the confidence of the input sequence x we used is defined as :

$$\mathbf{Confidence}(\mathbf{x}) = \operatorname{argmax} p(y|x)$$

3.3.2 Query Strategy based on NE features

NE features means Named Entity features. The main features for the NER task are identified based on the different possible combinations of available word and tag contexts. We use the following set of specific features, which is conducive to the improvement of the Tibetan named entity recognition performance.

a. Feature of tibetan person names

Tibetan person names could be divided into three catalogues: translation names and common names. The translation has special syllables and common tibetan person name has frequently used syllables. We collect 237 han surname as the feature of translation names. In common tibetan person name, we calculated the frequency based on 10460 Tibetan person name, and selected the top 97 as High frequency syllables .

In Tibetan, many words can indicate the boundary of names, such as ཀྲུའུ་ཞི། (chairman), དགེ་རྒན(teacher), བླ་མ(lamaism). These words are boundary word which has help for inspiration and instruction for person names. When these words appear in corpus, the credibility of name recognition will be improved.

b. Feature of Tibetan location names

Location names usually has particular syllables, such as རྫོང་(county), རི།(mountain). We collect 20 words as the feature of Tibetan location names.

c. Feature of Tibetan orgnization names

The feature of orgnization names and location names is practically identical. We collect 24 words as the feature of Tibetan orgnization names, inculding སློབ་གྲྭ(school), དངུལ་ཁང་།(bank).

4. Experiment and Discussion

4.1 Experiment design

In iterative development cycles, we select Top 10 sentences in each iteration. We test three different active learning methods: Random selection, Confidence-based Query Strategy, NE feature-based Query Strategy.

The result of random selection is the baseline in our experiment.

We use F1 to evaluate the performance of each graininess, which are very common in NLP evaluation.

P= (number of correctly identified NE)/(number of identified NE)

R= (number of correctly identified NE)/(number of all NE)

F1=(2*P*R)/(P+R)

4.2 Experiment data

We conducted our active learning experiments under Tibetan language. For our empirical evaluation, we used the training data and test data from four sites, include People's Network （Tibetan version）, Aba News Network, Tibet News Network, The Voice of America(Tibetan version). We marked the person name(PER), location names(LOC) and orgnization names (ORG) with a part of data,as labelled train data set and test data set, the remain corpus as unlabeled data set. There are about 7,000 sentences. Some statistics of training, development and test data are presented in Table 1.

	sentences	PER	LOC	ORG
Labelled train data set	249	231	164	76
Unlabeled data set	7269	-	-	-
Test data set	246	112	147	165

Table 1: Data source

4.3 Results and Analyses

The initial NER module gets an F-score of 10.7, while the train set contains only 249 sentences. We plotted the learning curves for the different query strategies.

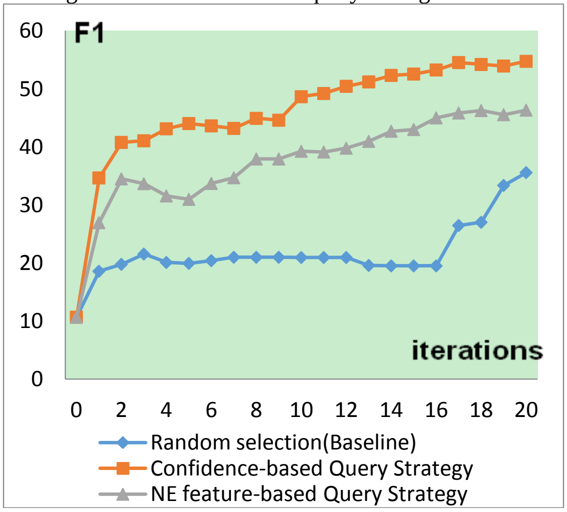

Figure 2: Comparison of active methods

The curves in Figure 2 show the relative performance. The F1 increases along with the number of selected sentences. The results suggest that both active learning methods consistently outperform the random selection.

The amplitude of variation in Random selection is irregular. After 16 iterations, the F1 is increasing. It says it is impossible to be sure about the value of the sentence we choose in Random selection.

The confidence-based query strategy has improved performance after each iteration. By comparison test, the strategy is better than random selection. The F1 has shot up by far more than the other methods. Under the same iterative numbers, the F1 increases from 10.7 to more than 54. The effect for the first two iterations is notable. After iterations, F1 is higher than the Random selection.

The NE feature-based query strategy also shows better result than Random selection. Although it is not as good as the confidence-based query strategy. Its dominance looks shaky. We think the reason for this is that named entity features we have collected are not enough, and there are still some exceptional circumstances.

5. Conclusion

Nowadays, the biggest cause for Tibetan Named Entity Recognition is the lack of training data. Because of the high cost and long time-consuming of tagging data, to get a lot of labeled data has been very difficult and expencive, and on the other hand, it is relatively easy to get a lot of unlabeled data. In this paper, we use active learning based on CRF to select useful data from a large number of unlabled corpus.The experiment shows that we can achieve better F1 by our mothods in the same iterative. We compared different active learning algorithms for Tibetan named entity recognition. Our results showed that active learning algorithms considerable performance improvements in reduced savings of annotation. In future research , we will investigate some new query strategies to get better effect.

6. Acknowledgement

The project was supported by Key Program of National Natural Science Foundation of China (Grant No. 61331013), Minzu University of China Tier-1 Universities and Tier-1 Subjects Graduate independent research projects (Grant No. 10301-0170040601-184).

7. Bibliographical References

Arkin M, Mahmut A, Hamdulla A. Person Name Recognition for Uyghur Using Condi-tional Random Fields[J]. International Journal of Computer Science Issues, 2013.

Chen Y, Lasko T A, Mei Q, et al. A study of active learning methods for named entity rec-ognition in clinical text[J]. Journal of biomedical informatics, 2015, 58: 11-18.

Culotta A, Kristjansson T, Mccallum A, et al. Corrective feedback and persistent learning for information extraction[J]. Artificial Intelligence, 2006, 170(14):1101-1122.

Cui B, Lin H, Yang Z. Uncertainty sampling-based active learning for protein–protein in-teraction extraction from biomedical literature[J]. Expert systems with Applications, 2009, 36(7): 10344-10350.

Dasgupta S, Hsu D. Hierarchical sampling for active learning[C]//Proceedings of the 25th international conference on Machine learning. ACM, 2008: 208-215.

Dou R, Jia YJ, Huang W.Automatic recognition of tibetan name with the combination of statistics and regular. Journal of Changchun Institute of Technology (Social Science Edition), 11 (2) 113-115.,2010.2:113-115.

Ekbal A, Saha S, Sikdar U K. On active annotation for named entity recognition[J]. Inter-national Journal of Machine Learning and Cybernetics, 2016, 7(4): 623-640.

Hua Q, Jiang W, Zhao H, et al. Tibetan name entity recognition with perceptron model[J]. Computer Engineering & Applications, 2014, 50(15):172-176.

Jia Y, Li Y, Zong C, et al. A Hybrid Approach Using Maximum Entropy Model and Con-ditional Random Fields to Identify Tibetan Person Names[J]. Himalayan Linguistics, 2016, 15(1).

Jia YJ, Yachao L I, Zong C, et al. A Hybrid Approach to Tibetan Person Name Identification by Maximum Entropy Model and Conditional Random Fields[J]. Journal of Chinese Information Processing, 2014.

Kang CJ, Long C, Jiang D. Tibetan names recognition research based on CRF[J]. Computer Engineering and Applications, 2015.

Kim S, Song Y, Kim K, et al. Mmr-based active machine learning for bio named entity recognition[C] // Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers. Association for Computational Linguistics, 2006: 69-72.

Mo H M, Nwet K T, Soe K M. CRF-Based Named Entity Recognition for Myanmar Lan-guage[J]. 2016.

Olsson F. A literature survey of active machine learning in the context of natural language processing[J]. 2009.

Rubens N, Elahi M, Sugiyama M, et al. Active learning in recommender sys-tems[M]//Recommender systems handbook. Springer US, 2015: 809-846.

Saha S, Ekbal A, Verma M, et al. Active learning technique for biomedical named entity extraction[C]//Proceedings of the International Conference on Advances in Computing, Communications and Informatics. ACM, 2012: 835-841.

Settles B. Active learning literature survey[J]. University of Wisconsin, Madison, 2010, 52(55-66): 11.

Sun Y, Yan X, Zhao X, et al. Research on automatic recognition of Tibetan personal names based on multi-features[C]// International Conference on Natural Language Processing and Knowledge Engineering. IEEE, 2010:1-5.

Tomanek K, Hahn U. Reducing class imbalance during active learning for named entity annotation[C]// Proceedings of the fifth international conference on Knowledge capture. ACM, 2009: 105-112.

Tomanek K, Laws F, Hahn U, et al. On proper unit selection in active learning: co-selection effects for named entity recognition[C]//Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing. Association for Computational Linguistics, 2009: 9-17.

Yao L, Sun C, Li S, et al. CRF-based active learning for Chinese named entity recogni-tion[C]//Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on. IEEE, 2009: 1557-1561.

Yu HZ, Jiang T, Ma N.Named Entity Recognition for Tibetan Texts[J].Lecture Notes in Engineering and Computer Science,2010,2180.

Zhu J, Li T, Liu S. Research on Tibetan name recognition technology under CRF[J]. Journal of Nanjing University, 2016.

Phonetically Based Extraction of Japanese Synonyms from Rakuten Ichiba's Item Titles

Ohnmar Htun[a], Koji Murakami[b], Yu Hirate[a]
[a]Rakuten Institute of Technology Tokyo, Rakuten Inc., Tokyo, Japan
ohnmar.htun@rakuten.com
yu.hirate@rakuten.com
[b]Rakuten Institute of Technology New York, Rakuten USA, Inc., New York, USA
koji.murakami@rakuten.com

Abstract

This paper presents a method for the phonetically based extraction of Japanese synonyms from item titles of Rakuten Ichiba. In general, synonyms are words with the same or similar meaning in a semantic sense; however, we focus here on those synonyms which appear as transliterations between English and Japanese, using Katakana, Hiragana, Kanji and a mixture of these scripts. The method consists of three parts: generation of the candidate word pairs using phrase detection (collocation) at the preprocessing stage; mapping similar sounds using Soundex and a cross-language sound group; measuring the similarity based on the Levenshtein and stochastic distances; and ranking the synonym pairs using fuzzy matching in the post-processing stage. We carry out two experiments based on two different sound mapping datasets, each of which measures the similarity scores from two different algorithms. The results from the baseline and cross-language models achieve precision values of 0.9208 and 0.9983, respectively. Our method is applicable to various fields of linguistic research, for example building a thesaurus/new name entity lookup for a search engine, machine translation and natural language generation, and improving output of voice recognition systems.

Keywords: Japanese synonym, transliteration mining, phonetic similarity, information retrieval

1. Introduction

Due to linguistic borrowing between languages, phonetic similarities can be found within a language (i.e., transcription) or between two or more languages (i.e., transliteration). In Japanese, Katakana is used to express sound effects and transliterated foreign words using Japanese pronunciation rules and syllables. The ending of words is therefore quite different from the original pronunciation. Fashion-related words are mostly constructed using foreign language words, for examples, "Lounge Style|ラウンジスタイル [RAUNJISUTAIRU]”， “Glenfield|グレンフィールド [GURENFIRUDO]”， and "Insignia Dress|インシグニアドレス [INSHIGUNIADORESU]".

Typically, synonyms are words with the same or similar meaning in a semantic sense, and can be easily found in a thesaurus. However, synonyms in Japanese can be found not only as semantically relevant words, but also as words that are phonetically equivalent across languages. For example, "basket" in English can be translated into Japanese as 籠 [KAGO], 篭 [KAGO], or transliterated as バスケット [BASUKETTO] by adopting sounds directly from the source language; this is also known as a "Loanword" or "Transliterated word". Newly created consumer products and services are being introduced to offline marketplaces and online digital market spaces on a daily basis, and many loanwords have been created as synonyms for consumer products in Japanese. In fact, query expansions in E-commerce search engines require the construction of sets of these synonymous names for concepts. The motivation for this work is to extract new synonym pairs from item-title phrases in the ladies' fashion database of Rakuten Ichiba (楽天市場) [1] to enhance the vocabularies of synonym dictionary in the search platform development.

In this work, we focus on extracting synonyms appearing as transliterations between English and Japanese, using Katakana, Hiragana and Kanji or a mixture of these scripts. The method presented here is an extension of prior research (Htun et al., 2011; Htun et al., 2012; Finch et al., 2012). The current approach is slightly different from previous studies; rather than bilingual pairs, the format of the test datasets contains long phrases with mixed encoding such as Latin alphabets, Japanese scripts, symbols and other annotated formats (e.g., date & measurement). The Gensim phrases (collocation) detection module (Mikolov T et al., 2013) is used to generate the candidate pairs in the preprocessing stage. The process of mapping sound uses Soundex (SDX) and cross-language sound grouping (CLSG). When measuring similarity, the Levenshtein distance (LD) algorithm (Levenshtein, 1966) is used to measure the CLSG directly, and each edit operation has a weight of one. The stochastic distance (SD) model (Ristad et al., 1998; Sajjad et al., 2012; Htun et al., 2012) is used to adjust the training parameters and iterations. The addition of a post-processing step with fuzzy matching[2] helps in extracting the synonyms accurately. The experiments generated two results since we constructed two models using baseline Soundex training (SDX-SD) and cross-language phonetic training (CLSG-SD). Our testbed contains 139,493 synonym pairs in the training data and 4,178,660 candidate pairs in the testing data. The results from baseline and cross-language models achieved a precision of 0.9208 and 0.9983 respectively.

The main contribution of this paper is the demonstration of a novel practical method by applying it to a real business support system; it is also applicable to various linguistic research studies, for example building a thesaurus/new name entity lookup for a search engine, machine translation and natural language generation, or improving the output of a voice recognition system. The remainder of the paper is organized as follows: in Section 2, we review prior

[1] https://www.rakuten.co.jp/

[2] https://pypi.python.org/pypi/fuzzywuzzy

research; Section 3 presents our methodology; Section 4 describes the experiments; Section 5 provides experimental data; Section 6 presents the results; Section 7 gives a short evaluation and discussion of the results obtained in the previous section; and Section 8 concludes this work.

2. Related Work

Earlier studies of phonetically based Japanese synonym extraction are reported by Tsuji et al. (2002). These authors manually construct transliteration rules between French and Japanese, and between English-Japanese. Katakana words convert into mora units[3], then match the character level between Japanese and French, and rank the pairs based on their frequency. They apply a string matching algorithm to find the longest common subsequence and use Dice to extract the word from the French part of corpora. However, the result achieves a precision of only 80% and a recall of 20%, the amount of the test data is very small.

A technique similar to phonetic matching has been applied to Japanese search engines using the PostgreSQL open source database by Yusukawa et al. (2012). They develop a sound grouping based on the similarity of Japanese speech sounds, and matching based on morphological analysis (MeCab[4]); they then extract terms from the document using Indri[5] and apply the Fuzzy string-matching function of PostgreSQL[6]. Using this method, they extract 84 million terms from the 67 million Japanese documents in the ClueWeb09-JA [7]collection. This work integrates an internal module of jpfuzzystrmatch into PostgreSQL. However, it suffers from an excessive generation of matches (i.e., both correct and incorrect).

Another approach to generating a large list of technical transliterated terms between Japanese and English employs a function of phrase-based statistical machine translation (PBSMT) function from Moses (Koehn et al., 2007). This is used to train a bilingual dictionary (Katakana-English) and aligned bilingual pairs (Japanese-English) using Wikipedia article titles (69,000 pair in total), and is tested with a large amount of data (24 million parallel title pairs). This method generates 7 million phrase pairs (Katakana-English) with high precision and recall, they consider to generate transliteration pairs from non-parallel data.

Prior research by Htun et al. (2012) and Finch et al. (2012) has been extended by adding a new approach (word reordering) to the joint process of transliteration and translation pairs (Finch et al., 2017) for mining bilingual lexicons from pairs of parallel short word sequences. They use four methods: the GIZA++ alignment tool (Och and Nay, 2003); the joint length base measure; stochastic edit distance based Dirichlet process model; and the stochastic edit distance base Dirichlet process model with word reordering. These are tested and evaluated using bilingual Wikipedia article titles in English-Japanese (137,780) and English-Chinese (192,407). However, this new approach achieves an F-score of only 0.898 for English-Japanese and 0.82 in English-Chinese, and the computational cost is excessively high. Our model uses only SD with a noise model (Htun et al., 2012) based on a single-word, and our current approach allows model learning of one or more words.

A variety of approaches have been proposed to extract Japanese-English transliterated pairs, most of which attempt to extract pairs from the bilingual corpora using different measures or learning algorithms. In recent years, the most popular word embedding model, Word2Vec (Mikolov et al., 2013a, 2013b), has enabled researchers to estimate the representations of words, as in the famous example: "King – Man + Woman = Queen". However, this representation cannot identify whether the words are similar to or different from each other in terms of pronunciation. Our approach uses phrase detection (collocation) to generate the candidate pairs. This approach gives a reduction in the computational cost of pairing and adds phonological knowledge support to the LD and SD model similarity scores. In post-processing, fuzzy partial matching eliminates duplicated extended pairs with the same sound. The experimental results show that our CLSG-SD model achieves a precision of more than 0.99, a significant improvement over previously proposed models (Htun et al., 2012).

3. Methodology

Our methodology consists of three steps:

- □ preprocessing;
- □ measurement of phonetic similarity; and
- □ post processing.

Figure 1 gives an overview of this methodology.

3.1 Preprocessing

3.1.1 Removing Abbreviations

We first clean abbreviations and formatted segments in the title strings using regular expression processing.

3.1.2 Parsing with MeCab

The cleaned strings are parsed using MeCab[8] for word segmenting, POS tagging and elimination of some unnecessary segments. (e.g., a segment "ので" in feature of "助詞,接続助詞/particle, connecting particle").

3.1.2 Pairing Using Phrase (collocation) Detection

The Phrases module in genism (Mikolov et al., 2013a, 2013b) has two basic steps.

- □ Collection of the word and word bigram frequencies, using a corpus of documents. This is referred to as training the model.
- □ Use of the trained model to detect phrases in the corpus. The detected phrase will merge with neighboring words if it is evaluated as being part of a collocation.

Trigrams use phrases transformed into bigrams as input, and we iterate the two steps above. We generate phrases based on bigram with the minimum count (i.e., min_count)

[3] https://en.wikipedia.org/wiki/Mora_(linguistics)
[4] https://github.com/taku910/mecab
[5] https://www.lemurproject.org/lemur/indexing.php
[6] https://www.postgresql.org/
[7] http://lemurproject.org/clueweb09.php/
[8] https://github.com/neologd/mecab-ipadic-neologd/blob/master/README.ja.md

set to one and the threshold set to nine. The trigram counts use the default parameters.

3.2 Measurement of Phonetic Similarity

3.2.1 Romanization and Simplification of Sounds

Our method involves only the measurement of phonetic strings. Non-Latin language scripts are therefore first converted into Romanized versions. We utilize various Japanese Romanization converters from the Python library, such as jaconv[9], romkan 0.2.1[10], and jProcessing 0.1[11]. The next step, simplifying sounds, has two stages. The first simplification corresponds to the native phoneme of each language. For example, gya[ギャ] is simplified as 'g', tsu[つ] is simplified as 'S' in Japanese, and 'sh' is simplified as 'S' in English. In the second step, we simplify this again using SDX and CLSG (Kodama, 2010; Htun et al., 2011; Htun et al., 2012).

3.2.2 Measuring Similarity

Levenshtein Distance

The LD (Levenshtein, 1966) is a dissimilarity measure between two strings. It is the minimum number of character edits required to transform one string into the other, using the edit operations of insertion, deletion, or substitution of a single character. The editing cost for each operation set is one, and the LD is calculated as follows:

$$LD = I + D + S \qquad (1)$$

where I = the number of insertions
D = the number of deletions
S = the number of substitutions

The LD is normalized, denoted here by LDn, and defined as follows:

$$LDn = 1 - \left(\frac{LD}{(L1 + L2)}\right) \qquad (2)$$

where L1 and L2 are the lengths of converted strings from the sound simplification process. LD_n lies in the range $0 \le LD_n \le 1$. We refer to this score as the LD similarity result.

Stochastic Distance

The SD is an unsupervised generative model (Ristad et al., 1998; Sajjad et al., 2012) that can assign a joint probability to a pair of strings using the probabilities of edit operations. An edit cost (P_j) is calculated by applying the negative logarithm to the joint probability of an edit (e) as given below:

$$P_j = -Log(P(e)) \qquad (3)$$

Exponentially many edit sequences may be generated, and this increases the probability of the entire string pair P(X,Y). The edit distance is defined as $d_s(X,Y)$ and is calculated by summing the derivation probabilities over all paths as follows:

$$d_s(X,Y) = \sum_{s \epsilon Z} \sum_{j \epsilon s} P_j \qquad (4)$$

$Z = \{s_1, s_2, s_3, ..., s_i\}$ is the set of all edit operation sequences that are generated between strings X and Y.

An edit is represented by j, and $s = (j_1, j_2, j_3, ..., j_n)$ denotes a sequence of edits (an edit path).

The edit costs are learned using the expectation maximization (EM) algorithm, which involves a forward-backward dynamic programing technique. The SD learns using data with both transliteration and non-transliteration, and has two sub-models: transliteration (clean model), which assigns a high probability, and non-transliteration (noise model), which assigns a low probability. The full SD model is an interpolation of both models.

$$P(X,Y) = (1-\lambda)P_t(X,Y) + \lambda P_n(X,Y) \qquad (5)$$

where λ is the prior probability of the data being noise (a non-transliteration pair), P_t is the probability of the clean model, and P_n is the probability of the noise model.

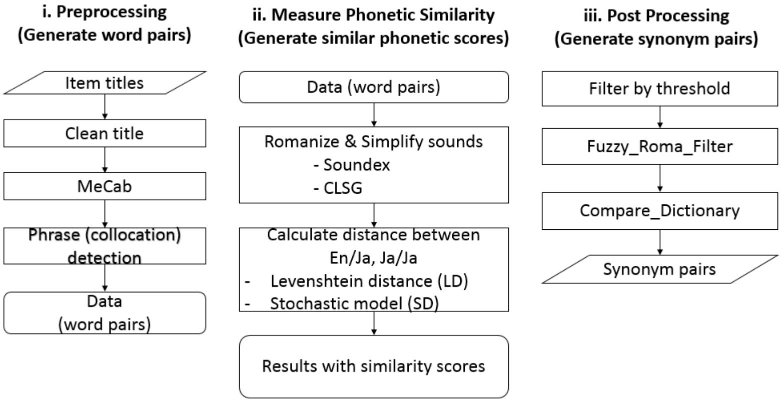

Figure 1: Overview of the methodology

[9] https://pypi.python.org/pypi/jaconv/
[10] https://pypi.python.org/pypi/romkan
[11] https://pypi.python.org/pypi/jProcessing/0.1

3.3 Post-processing

3.3.1 Filter by Thresholds

Thresholding is commonly used in information retrieval (IR) analysis. It is a procedure similar to clustering to assign a similarity score to a class indicating whether or not the score is greater than a predefined threshold. The performance of IR algorithms depends on the output quality of the thresholding process. For example, we assign a threshold value (T) to SD scores as: non-synonym > T ≥ synonym. We use joint thresholds (both LD and SD) in each experimental result.

3.3.2 Fuzzy Roma Filter

To eliminate pairs with similar sounds and meaning with one or more additional characters (known as pairing error), the fuzzy ratio function is used to rank these kinds of similar strings and to extract the top-ranked string.

3.3.3 Dictionary Comparison

The main objective of this stage is to extract new synonym pairs which are not included in existing dictionaries. This function involves only straightforward matching with synonym pairs from existing dictionaries. Finally, the new synonym pairs are extracted.

4. Experiments

The experiments were carried out to measure phonetic similarity using two methods on two different phonetic coding datasets, giving a total of four experimental conditions as shown below:

Experiment - I		**Experiment -II**	
SDX Grouping Data		CLSG Grouping Data	
Levenshtein	Stochastic	Levenshtein	Stochastic

Table 1: Set of experiments

Experiment I involves two algorithms using SDX, and the baseline measurements are compared to the results from Experiment II.

Soundex:
The Romanized candidate pairs are converted to a four-character code that is based on the six-articulation group. For example, the candidate pair "bamboo grass|バンブーグラス" is converted into SDX coding as "B512|B512".

Cross-Language Phonetic Grouping:
The CLSG approach is an extension of Soundex, and focuses on finding similar-sounding text between English and a group of Asian languages: Indonesian, Japanese, Korean, Malay, Myanmar, Thai, and Vietnamese. This experiment used CLSG version 1. For example, the candidate pair "bamboo grass|バンブーグラス" is converted into CLSG coding as "191574|191574".

Levenshtein Distance:
In (Htun et al., 2011), a variable weight in substitution operation sets 0.5 if the relation of phonetic coding characters belongs to the same place of articulation and manner; however, it sets 1 if it is not in the same place of articulation and manner. In this experiment, we apply 1 for each operation (i.e., insertion, deletion, and substitution) and measure directly to the phonetic coding converted strings.

Stochastic Distance:
The model was trained in a completely unsupervised way. The average training time was about two hours for 242,207 pairs, using 400 training iterations. Testing time was mostly less than one minute in all cases, from the minimum 55,892 training pairs to the maximum of 1,188,291. Training and testing data should use the phonetic coding; otherwise, the model cannot learn from the testing data. The SD function returns a probability score between 0 and 1.

Threshold and Filtering:
We used a joint threshold to filter out non-phonetic synonym pairs. In the baseline experiment, we allocated joint thresholds of a SDX-LD similarity score and a SDX-SD probability score of 0.875 and 0.9999 respectively. In the same way, the experiment using CLSG data applied a joint threshold of a CLSG-LD similarity score and CLSG-SD probability score of 0.85 and 0.9999 respectively.

5. Data

5.1 Training Data

The training data contained 242,207 synonym pairs of Japanese-English transliterations and Japanese-Japanese transcriptions, taken from the existing thesaurus dictionary and the Egi (RIT) transliteration dataset (2017). Training data was also required to clean unnecessary numerical characters, symbols, and so on. Some examples of source training data pairs (before cleaning and converting to phonetic transcriptions) are given in Table 2.

Synonym-1	**Synonym-2**
黒糖クルミ	黒糖くるみ
カツウラ化粧品	かつ化粧品
黒胡椒黒胡麻ペースト	黒ごまペースト
TIMETIMER	タイムタイマー
TIME VOYAGER	タイムボイジャー
ロストボール	ろすとボール
mickeycandybowl!	ミッキーキャンディーボール
ベッキー♪#	ベッキー
任天堂 wifi	ニンテンドーwi-fi

Table 2: Examples of source training data

Figure 2 shows the statistics for the types of synonym pairs. The greatest number of synonym types was English-Katakana transliteration pairs, with 171,867 in total. The lowest number of synonym types were English-Hiragana and English-Kanji with 313 and 328 receptively.

5.2 Test Data

The test dataset was extracted from titles of Rakuten Ichiba women's fashion items, and contained a total of 5,821,560 titles in 12 sub-categories. Following the process of pairing, 4,178,660 candidate pairs were generated. Table 3 presents statistics for the number of titles and generated candidate pairs in each sub-category.

	Sub-category women’s Fashion	**# of titles**	**# of candidate pairs**
1	Tops	1,831,078	1,188,291
2	Dresses	172,441	162,482
3	Outerwear	531,059	446,724
4	Bottoms	989,564	687,201
5	Other Fashion	187,290	148,188
6	Others	431,931	217,840
7	Suits	54,302	57,268
8	Kimonos	609689	440,741
9	One Piece Dresses	714,313	528,371
10	Costumes	149,469	154,802
11	Swimwear	111,543	90,860
12	All-in-One	38,881	55,892

Table 3: Number of titles and candidate pairs generated in each sub-category by the CLSG test

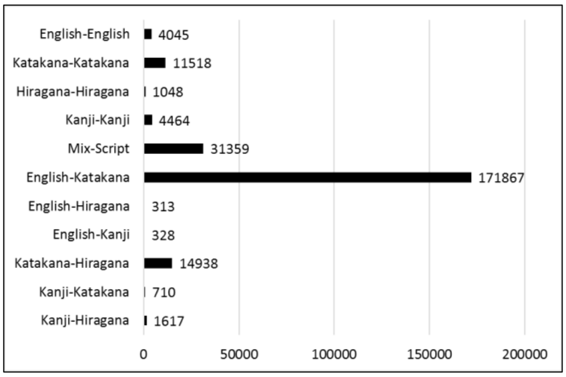

Figure 2: Type of synonyms in training data

	Experiment I		**Experiment II**	
Synonym pair	**SDX_LD**	**SDX_SD**	**CLSG_LD**	**CLSG_SD**
mawaru penguindrum|輪るピングドラム	1	0.999997	0.947368	1
senbonzakura|千本桜衣装	1	0.999996	0.875	0.999969
rage burst|レイジバースト	0.875	0.999905	0.909091	0.999962
parasite chest|パラサイトチェスト	1	0.999996	0.866667	0.999976
bone princess|ボーンプリンセス	1	0.999998	1	1
ensemble star fine|あんさんぶるスターズ	0.875	0.999874	0.85	0.999992
touka gettan|桃華月憚	1	0.999994	0.909091	0.999949
durarara|デュラララ	1	0.999997	1	0.999941

Table 4: Examples of phonetic similarity scores from the results of Experiments I and II

		Experiment I (SDX)			**Experiment II (CLSG)**		
	Subcategory	**Extracted pairs**	**Recall**	**Precision**	**Extracted pairs**	**Recall**	**Precision**
1	Tops	7,104	0.6861	0.90	5,649	0.7045	0.99
2	Dresses	466	0.7036	0.97	400	0.7000	1.00
3	Outerwear	4,027	0.6907	0.97	3,274	0.7045	0.99
4	Bottoms	4,720	0.8875	0.90	3,949	0.7000	1.00
5	Other Fashion	498	0.7017	0.88	385	0.7000	1.00
6	Others	1,230	0.7098	0.87	1,011	0.7000	1.00
7	Suits	533	0.7000	1.00	423	0.7000	1.00
8	Kimonos	314	0.6944	0.90	211	0.7000	1.00
9	One Piece Dresses	3,648	0.6925	0.87	3,019	0.7000	1.00
10	Costumes	443	0.6995	0.94	357	0.7000	1.00
11	Swimwear	308	0.6896	0.91	261	0.7000	1.00
12	All-in-One	553	0.7074	0.94	474	0.7000	1.00

Table 5: Number of extracted synonym pairs and precision of random 100 samples in each sub-category

6. Results

Several examples of phonetic similarity scores from the results of Experiment I and II are shown in Table 4. The scores returned by each method are scaled to the range [0,1]. We used the metrics of precision and recall, and Table 5 shows the performance of both LD and SD for each experiment. We used a phonetic similarity measure technique to extract synonym candidates, and extracted 23,844 pairs of synonyms for the baseline, with an average precision of 0.9208, and about 19,413 pairs of synonyms in Experiment II with a high precision of 0.9983 on average. In each experiment, we applied a joint threshold of 0.875 for SDX-LD and 0.9999 for SDX-SD for the baseline Experiment I, and a joint threshold of 0.85 for CLSG-LD and 0.9999 for CLSG-SD in Experiment II.

7. Discussion

The proposed methodology aims to produce synonym word pairs that are not found in the existing dictionaries of Rakuten Ichiba. We therefore focused on extracting as many synonyms as possible, whereas the results should exclude the synonyms from existing dictionaries.

Paring Words/Phrases

In our test data, item titles were mix-encoding strings which form pairs of English and Japanese words or phrases. We developed an approach utilizing the phrase detection function of the Genism library to pair words or phrases (Mikolov et al., 2013a, 2013b). This technique greatly reduced the computational cost of generating all possible pairs in each test category dataset.

Phonetic Coding

Although the various language scripts are written in Latin/Romanized scripts, the spelling does not always correspond directly to the pronunciation. Because loanwords are generally written in Katakana/Romaji and are pronounced using Japanese pronunciation rules and Japanese syllables, there may be many variations in spelling for the same transliteration. In this experiment, we focused on extracting not only transliteration between English and Katakana, but also between English and Romaji, Hiragana and Kanji. A novel approach based on CLSG helped to increase the precision and reduce the parameter of the learning process.

Measuring/Learning

Normalizing the LD value makes it easy to determine a threshold of best-N extraction from the results. LD can be applied rapidly to diverse information retrieval (IR) tasks. In our previous work, SD learned a one-to-one form of bilingual word pairs (e.g., platinum|プラチナ), whereas now it can learn phrases/segments, for example “v-neck pullover deck shirts|vネックプルオーバーデッキシャツ”.

Thresholding

The allocation of a threshold is a key to differentiate synonym and non-synonym pairs. In this experiment, we manually set a reasonable value for the threshold for each method, and then evaluated the precision of a randomly selected 100 synonym pairs in the final step (i.e., after excluding synonym pairs from the existing dictionaries). Although the use of joint thresholds in each experiment optimized the synonym extraction task, the allocation of thresholds had to be done manually. Automatic allocation should therefore be considered in the future.

Fuzzy Ranking

Due to frequent co-occurrences (words/phrases) in the paring process, some incorrect pairs appeared as one or more unnecessary characters in addition to the words. For example, if we applied an individual threshold of 0.9999 to CLSG-SD, this kind of error could be avoided; otherwise, the fuzzy score can be satisfied to eliminate these incorrect pairs (See Table 6).

Synonym pairs	**CLSG-SD**	**Fuzzy rank**
dub_collection|ダブコレクションリング	0.997029	41
dub_collection|ダブコレクション	**0.999999**	**48**
dub_collection|ダブコレクションダブ	0.999389	42
dub_collection|ダブコレクションレディース	0.997871	39

Table 6: Examples of error pairs and scores in CLSG-SD and fuzzy ranking

Evaluation

For the evaluation, a sample of 100 synonym pairs from each category result was first randomly selected. There were 1,200 synonym pairs for 12 categories in Experiments I and II respectively. Then, each experiment sample set was annotated manually and used to calculate the precision and recall (See Table 5). The results of Experiment II showed improved performance for the CLSG coded dataset over Experiment I (i.e., the baseline), which used the SDX coded dataset (See Figure 3).

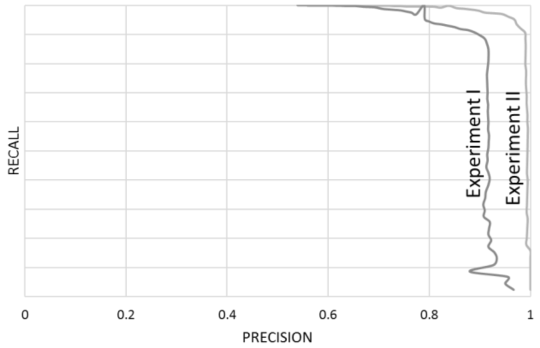

Figure 3: Performance comparison for Experiments I & II

8. Conclusion

We present here a practically oriented approach for the extraction of Japanese synonyms based on phonetic similarity, with high precision. Our test datasets are not bilingual pairs, and the generation of candidate pairs therefore posed a challenge at the early stages, since we do not want to omit any possible pairs in the generation process. Integration of the phrase detection module of genism reduced the computational cost and maximized the coverage of bilingual candidate pairs. However, SD learning improved from one-to-one word pairs to one-or-more phrases, and the probabilistic scores of synonyms were higher than in previous studies. In future work, we aim to investigate ways of optimizing the learning parameter of the SD model. Allocation of the thresholding process also requires improvement. Experiment II achieved high values for precision. In the future, we intend to develop a deep learning neural network model integrated with a phonetic concept to enhance the performance. We also aim to extend our system to extract the synonym pairs in other languages.

9. References

Finch, A., Harada, T., Tanaka-Ishii, K., and Eiichiro Sumita (2017). Inducing a Bilingual Lexicon from Short Parallel Multiword Sequences. ACM Trans. Asian Low-Resour. Lang. Inf. Process., 16(3), Article 15DOI: https://doi.org/10.1145/3003726

Finch, A. M., Htun, O., and Sumia, E. (2012). The NICT translation system for IWSLT 2012. In Proceedings of the 7th International Workshop on Spoken Language Translation (IWSLT"12). 121–125.

Ristad, E. S., and Yianilos, P N. (1998). Learning string edit distance. IEEE Transactions on Pattern Recognition and Machine Intelligence, 20(5):522–532.

Och, F. J., and Ney, H. (2003). A systematic comparison of various statistical alignment models. Computational Linguistics, 29(1):19–51.

Sajjad, H., Fraser, A., and Schmid, H. (2012). A statistical model for unsupervised and semisupervised transliteration mining. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, 1:469–477. http://www.aclweb.org/anthology/P12-1049

Richardson, R., Nakazawa, T., and Kurohashi, S. (2014). Bilingual dictionary construction with transliteration filtering. In Proceedings of LREC 2014, Ninth International Conference on Language Resources and Evaluation, 1013–1017

Tsuji, K., Daille, B., and Kageura, K. (2002). Extracting French-Japanese word pairs from bilingual corpora based on transliteration rule. In Proceedings of 3rd LREC, pp. 499–502.

Yusukawa, M., Culpepper, J. S., and Scholer, F. (2012). Phonetic matching in Japanese. In Proceedings of SIGIR 2012 Workshop on Open Source Information Retrieval (OSIR2012), Portland, Oregon, USA, 68–71. http://opensearchlab.otago.ac.nz/

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a) Efficient estimation of word representations in vector space. ArXiv13013781Cs. Available: http://arxiv.org/abs/1301.3781, accessed 11 June 2017

Mikolov, T., Yih, W.-T., and Zweig, G. (2013b). Linguistic regularities in continuous space word representations. HLT-NAACL.

Htun, O., Shigeaki, K., and Mikami, Y. (2011). Cross-Language Phonetic Similarity Measure on Terms Appeared in Asian Language. International Journal of Intelligent Information Processing, 2(2).

Htun, O., Finch, A., Sumita, E., and Mikami, Y. (2012). Improving transliteration mining by integrating expert knowledge with statistical approaches. International Journal of Computer Applications, 58.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin 4, A., and Herbst, E. (2007). Moses: Open Source Toolkit for Statistical Machine Translation. In ACL 2007.

Shigeaki, K (2010). String edit distance for computing phonological similarity between words. In Proceedings of the International Symposium on Global Multidisciplinary Engineering.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. Journal of Soviet Physics Doklady, 10(8):707–709.

Wu, X. (2013). Mining Japanese compound words and their pronunciations from web pages and tweets. In Proceedings of the International Joint Conference on Natural Language Processing, Nagoya, Japan, 14-18 October 2013, 849–853.

Xianchao Wu, "Mining Japanese Compound Words and Their Pronunciations from Web Pages and Tweets", In proceedings of International Joint Conference on Natural Language Processing, pages 849–853, Nagoya, Japan, 14-18 October 2013.

Low-Resource Neural Machine Translation with Transfer Learning

Tao Feng[1,2], **Miao Li**[1], **Lei Chen**[1]
[1]Institute of Intelligent Machines, Chinese Academy of Science, Hefei, China
[2]University of Science and Technology of China, Hefei, China
ft2016@mail.ustc.edu.cn, {mli, chenlei}@iim.ac.cn

Abstract

Neural machine translation has achieved great success under a great deal of bilingual corpora in the past few years. However, it does not work well for low-resource language pairs. In order to solve this problem, we present a transfer learning method which can improve the BLEU scores of the low-resource machine translation. First, we exploit encoder-decoder framework with attention mechanism to train one neural machine translation model with large language pairs, and then employ some parameters of the trained model to initialize another neural machine translation model with less bilingual parallel corpora. Our experiments demonstrate that the proposed method can achieve the excellent performance on low-resource machine translation by weight adjustment and retraining. On the IWSLT2015 Vietnamese-English translation task, our model can improve the translation quality by an average of 1.55 BLEU scores. Besides, we can also get the increase of 0.99 BLEU scores when translating from Mongolian to Chinese. Finally, we analyze the results of experiments and summarize our contribution.

Keywords: Low-resource, Neural machine trannlation, Transfer learning

1. Introduction

Machine translation is one of the most important research field of artificial intelligence and natural language processing, which explores how to use computers to translate automatically between natural languages. In recent years, end-to-end neural machine translation has developed rapidly. The key idea of end-to-end neural machine translation is to achieve automatic translation between natural languages through neural networks. Compared with statistical machine translation, the quality of translation has been significantly improved. In a variety of languages pairs, the performance of neural machine translation has gradually surpassed phrase-based statistical machine translation. (Junczys-Dowmunt et al, 2016) provided comparison of translation quality for phrase-based statistical machine translation and neural machine translation across 30 translation directions with United Nations Parallel Corpus v 1.0. The results showed that neural machine translation surpassed statistical machine translation in 27 languages pairs.

Encoder-decoder (Sutskever et al, 2014) is one of most commonly framework in neural machine translation. The main idea of the framework is to map the input sequence to a fixed-sized vector with encoder, and then map the vector to the target sequence with decoder. Compared to traditional statistical machine translation, encoder-decoder has two major advantages. First, the framework can learn features directly from raw data. The results show, encoder-decoder learns sensible phrase and sentence representations that are sensitive to word order and are relatively invariant to the active and the passive voice (Sutskever et al, 2014). Second, the framework effectively captures long-range dependencies based on long short-term memory networks (Hochreiter & Schmidhuber, 1997). Therefore, encoder-decoder framework can significantly improves the fluency and readability of the translation. However, encoder-decoder framework needs to map an input sentence of variable length into a fixed-dimensional vector representation, which poses a great challenge for the encoder to deal with long sentences. In order to solve this problem, (Bahdanau el al, 2014) proposed attention mechanism to dynamically computer the context of the source end. Attention mechanism changes the way of information transfer, and significantly improve the performance of neural machine translation. Therefore, the encoder-decoder framework with attention has become the mainstream method of the neural machine translation.

However, as a data-driven approach, the performance of neural machine translation is highly dependent on the size and the quality of parallel corpora. As is known to all, neural machine translation will fail when training data is not big enough (Koehn & Knowles, 2017). In some low-resource translation tasks, the performance of neural machine translation is severely reduced. However, the vast majority of the languages in the world lack large, high-quality parallel corpora (Artetxe et al, 2017). Therefore, research on low-resource neural machine translation is valuable.

In this paper, we propose a simple and effective method to alleviate this problem. First, we train one neural machine translation model with large parallel corpora, and then transfer some parameters of the trained model to initialize another neural machine translation model with less parallel corpora without changing neural network architecture. Whether it is a high-resource language pair or low-resource language pair, we use the encoder-decoder framework with long short-term memory units(LSTM). As illustrated in Figure 1, in the encoder-decoder framework, an encoder at the source compresses the source sentence into a semantic vector, and another decoder at the target side generates a sentence based on this vector. However, a potential issue of this encoder-decoder approach is that neural network needs to be able to compress all the information of a source sentence into a fixed-length vector. This may make it difficult for the neural network to cope with long sentences, especially those that are longer than sentences in the training corpus (Bahdanau et al, 2014). Therefore, we add global attention mechanism (Luong et al, 2015) for each target language. Attention mechanism can better solve the problem of long distance information transmission and significantly improve the performance of neural machine translation.

We follow the transfer learning method proposed by (Zoph et al, 2016). In their work, the high-resource language pair

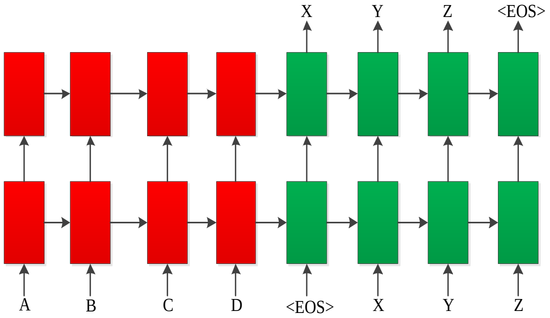

Figure 1: The encoder-decoder framework for neural machine translation. The framework learns to encode a variable-length sequence into a fixed-length vector representation and to decode a given fixed-length vector representation back into a variable-length sequence.

is called the parent model and the low-resource language pair is called child model. The parent model is first trained. Then the parameter values of child model are copied from the parent model and are fine-tuned. Comparison to their work—while our approach is similar in spirit to the model proposed by them, there are several key differences which reflect how we have simplified from the original model .

1. Both in high-resource language pairs and low-resource language pairs, they used uni-directional LSTM at the encoder end. However, we use bi-directional LSTM at the encoder because we would like the annotation of each source word to summarize not only the preceding words, but also the following words. So, our model works better than theirs on long sentence.

2. In their work, the target word embeddings of the child model are copied from the parent model and are frozen during fine-tuning because the target language is same in both parent model and child model. However, in our model, the target word embeddings of child model are initialized randomly and are constantly updated during training. The expression of language is not same in different domains. For example, the expression of sentence in the spoken corpus is more casual, while sentences in news corpus are more formally expressed. However, a word is characterized by the company it keeps (Harris, 1981), so the embedding of same word in different domains is not same. It is not a good choice to remain target word embedding of child model frozen.

2. Related work

Low-resource neural machine translation has attracted a lot of attention in recent years. The performance of neural machine translation is severely reduced when the parallel corpora is not enough. An effective way to alleviate this problem is to extend the scale of parallel corpora. (Sennrich et al, 2016) proposed a method to use the existing machine translation system to translate monolingual data and constructed dummy parallel corpora to relief the issue of lack of corpora. (Currey et al, 2017) utilized neural machine translation system to both translate source language text and copy target-language text, thereby exploiting monolingual corpora in the target language. Besides, for the low frequency words, (Fadaee et al, 2017) proposed a data augmentation method, which is also an effective way to extend the parallel corpora. Zero-resource neural machine translation is another way to deal effectively with insufficient resources, which is usually used in a pivot language. (Johnson et al, 2017) proposed a multilingual neural machine translation method. (Chen et al, 2017) presented a method for zero-resource neural machine translation by assuming that parallel sentences have close probabilities of generating a sentence in a third language called teacher-student framework. (Zheng et al, 2017) showed that maximum expected likelihood estimation can significantly improve the performance of zero-resource neural machine translation.

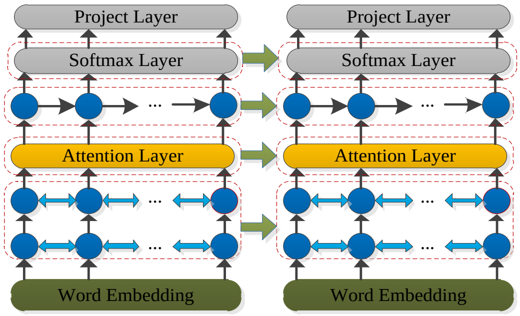

Figure 2: Aritecture of the proposed system. The left one is trained with high-resources, and then we transfer some parameters of trained model to initialize the right one. As is shown above, we transfer the parameters of all layers of the trained model except for the projection layer and word embedding layer.

3. Proposed Method

Figure 2 summarizes this general schema of the proposed system. This section describes the proposed neural machine translation with transfer learning. Section 3.1 first presents the architecture of the basic model, and section 3.2 then describes how to transfer parameters from high-resource language pairs to low-resource language pairs.

3.1 Basic models

Whether it is a high-resource language pair or low-resource language pair, the same neural network architecture is used. As shown in figure 1, the proposed system follows a standard encoder-decoder architecture with an attention mechanism.

In detail, we use a two-layers bi-directional RNN in the encoder, and another two-layers uni-directional RNN in the decoder. All the RNNs use LSTM cells with 1024 units, and the dimensionality of word embedding is set to 1024. As for attention mechanism, we use the global attention method proposed by (Luong et al, 2015). The model are trained using stochastic gradient descent with a minibatch size of 128 and a maximum sentence length of 50. We apply dropout (Gal et al, 2016) in high-resource language pair model with a probability of 0.2 and 0.5 in low-resource language pairs model. For all models, the learning rate decreases as the increase of the number of iterations. We decode using beam search on models with a beam size of 10. We initialize all of the parameters of network with the uniform distribution. We set the maximum value of the gradient to 5 in order to solve gradient explosion.

3.2 Transfer learning model

In short, transfer learning exploits knowledge from a learned task (source task) to improve the performance on a related task (target task), typically reducing the amount of required training data (Pan &Yang, 2010). Generally, the amount of data in the source task is sufficient, and the amount of data in the target area is small. Transfer learning needs to transfer the knowledge learned in the condition of sufficient data to the new environment with small amount of data. Traditional machine learning assumes that training data and testing data share same feature space and the same data distribution. When there is a difference in the data distribution between the training data and testing data, the results of predictive learner can be degraded (Shimodaira, 2000). However, transfer learning relaxes the limitation requirement, and applies the acquired knowledge to different but similar domains, which solves the problem of insufficient training data in the target domain. The transfer learning is usually divided into three types: instance-transfer, feature-representation-transfer, relational-knowledge-transfer.Transfer learning has been applied to many fields of the natural language processing, such as text categorization and machine translation. (Dai et al, 2007) proposed a co-clustering based classification algorithm to classify documents across different domains. (Long et al, 2010) propose Dual Transfer Learning method, which can improve the performance of classification accuracy.

In our paper, we translate Vietnamese into English with the help of French-English. First, we train French-English neural machine translation model, and then Vietnamese-English model is initialized with the parameters of the trained model. We just transfer some parameters to Vietnamese-English model, such as weights and biases of neural network, not all of them.

We follow the transfer learning method proposed by (Zoph et al, 2016). However, we have two improvements over their work. First, our model use bi-directional LSTM at the encoder because we would like each source word to summarize not only the preceding words, but also the following words. Second, we consider that the expression of language is not same in different domains. For example, the expression of sentence in the spoken corpus is more casual, while sentences in news corpus are more formally expressed. Moreover, a word is characterized by the company it keeps, so the embedding of same word in different domains is not same. Therefore, the target word embedding of Vietnamese-English model is initialized randomly instead of being copied from the French-English model. In addition, the projection layer of Vietnamese-English model can not be copied from French to English model, because target vocabulary of these two models is different. In order to verify the effectiveness of the method, we also translate Mongolian into Chinese with the help of English-Chinese.

4. Results and Analysis

4.1 Dataset details

As is shown in Table 1,Vietnamese-English corpora (133K sentence pairs, 2.7 million English words and 3.3 million Vietnamese words) is provided by IWSLT2015 and Mongolian-Chinese (67K sentence pairs, 848K Chinese words and 822K Mongolian words) is provided by CWMT 2009. We evaluate our approach on the French-English (2 million sentence pairs, 50 million English words and 52 million French words) translation task of the WMT14 workshop. And we get English-Chinese corpora (2 million sentence pairs, 22 million English words and 24 million Chinese words) from the WMT2017. We preserve casing for words and replace those whose frequencies are less than 5 by <unk>. As a result, our vocabulary sizes are 17K and 7.7K for English and Vietnamese respectively. And we report BLEU scores on tst2012 and tst2013.For Chinese-Mongolian corpora (67K sentence pairs, 849K Chinese words and 822K Mongolian words), we make the same treatment. Therefore, the vocabulary size of Chinese and Mongolian are 14K and 12K respectively.

Dataset		sentences	words
Fr-En	Frence	2m	52m
	English		50m
Vi-En	Vietnamese	133K	2.7m
	English		3.3m
En-Ch	Chinese	2m	24m
	English		22m
Mn-Ch	Mongolian	67K	822K
	Chinese		894K

Table 1 : Statistics of all datasets

Models	BLEU	
	tst2012	tst2013
Baseline	20.43	23.17
Ours	**21.86**	**24.83**
(Luong & Manning,.2015)	-	23.3

Table 2: The performance of the proposed method on IWSLT2015 Vietnamese to English tst2012 and tst2013 set.

Models	BLEU
Baseline	11.69
Ours	**12.68**

Table 3: The performance of the proposed method on CWMT2009 Mongolian to Chinese test set.

4.2 Results

The results of BLEU scores are presented in Table 2 and Table 3. The architecture of baseline system is similar to the one mentioned in section 3.1. However, in order to prevent overfitting, we use one-layer bi-directional LSTM in the encoder, with 512 cells at each layer and 512 dimensional word embeddings.

As it can be seen, the proposed transfer learning method obtains very competitive results considering that it was trained on nothing but low-resource corpora. Our model can reach 21.86 and 24.83 BLEU scores in Vietnamese-English tst2012 and tst2013 set respectively, we can also achieve 12.86 BLEU scores in Chinese-Mongolian test set, which is much stronger than the baseline system, with improvements of at least 6.9% in all cases, and up to 8.5% in some (e.g. from 11.69 to 12.68 BLEU scores in Mongolian to Chinese). As you can see from the results, the proposed method obtains substantial improvement over baseline system, indicating that transfer learning method is significantly effective. Therefore, our method can improve the performance of low-resource machine translations.

5. Conclusion and future works

In this paper, we propose a novel method to train low-resource neural machine translation system. First we utilize encoder-decoder framework with attention to train one neural machine translation with high-resource language pairs, and then transfer some parameters of the trained model to initialize another neural machine translation model with less bilingual parallel corpora.

The experiments show the effectiveness of our proposal, obtaining significant improvements in the BLEU scores over baseline system. Our model can improve the translation quality on the IWSLT2015 Vietnamese-English translation task. We can achieve 21.86 and 24.83 BLEU scores on Vietnamese-English tst2012 and tst2013 set respectively. Besides, the method in this paper is also effective for Mongolian-Chinese translation. And we can improve the performance CWMT2009 Mongolian-Chinese translation task by 0.99 BLEU scores.

In the future, we plan to combine unsupervised or semi-supervised methods with transfer learning approach. Besides, we will verify the method with more datasets from different domains.

6. Acknowledgments

The work is supported by the National Natural Science Foundation of China under No. 61502445, 61572462.

7. Bibliographical References

Junczys-Dowmunt, M., Dwojak, T., & Hoang, H. (2016). Is neural machine translation ready for deployment? a case study on 30 translation directions. arXiv preprint arXiv:1610.01108.

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *Computer Science.*

Philipp Koehn and Rebecca Knowles. (2017). Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation* (pp. 28–39)

Artetxe, M., Labaka, G., & Agirre, E. (2017). Learning bilingual word embeddings with (almost) no bilingual data.In *Meeting of the Association for Computational Linguistics* (pp.451-462).

Sennrich, R., Haddow, B., & Birch, A. (2015). Improving neural machine translation models with monolingual data. *Computer Science*.

Anna Currey, Antonio Barone and Kenneth Heafield. Copied monolingual data improves low-resource neural machine translation. (2017). In *Proceedings of the Conference on Machine Translation*(pp.148-156)

Fadaee, M., Bisazza, A., & Monz, C. (2017). Data augmentation for low-resource neural machine translation. In *Proceedings of 55th Annual Meetings of Association for Computational Linguistics*.

Johnson, M., Schuster, M., Le, Q. V., Krikun, M., Wu, Y., & Chen, Z., et al. (2016). Google's multilingual neural machine translation system: enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*(pp.339-351)

Chen, Y., Liu, Y., Cheng, Y., Li, V. O. K., Chen, Y., & Liu, Y., et al. (2017). A Teacher-Student Framework for Zero-Resource Neural Machine Translation. In *Meeting of the Association for Computational Linguistics* (pp.1925-1935).

Zheng, H., Cheng, Y., Liu, Y., Zheng, H., Cheng, Y., & Liu, Y., et al. (2017). Maximum Expected Likelihood Estimation for Zero-resource Neural Machine Translation. In *Twenty-Sixth International Joint Conference on Artificial Intelligence* (pp.4251-4257).

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Proceedings of the Neural Information Processing Systems.*

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735.

Zoph, B., Yuret, D., May, J., & Knight, K. (2016). Transfer learning for low-resource neural machine translation. In *Conference on Empirical Methods in Natural Language Processing*.

Harris, Z. S. (1981). Distributional Structure. *Papers on Syntax. Springer Netherlands*.

Luong, M. T., Pham, H., & Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. *Computer Science.*

Gal, Y., & Ghahramani, Z. (2015). A theoretically grounded application of dropout in recurrent neural networks. *Statistics*, 285-290.

Shimodaira, H. (2000). Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning & Inference, 90*(2), 227-244.

Dai, W., Xue, G. R., Yang, Q., & Yu, Y. (2007). Co-clustering based classification for out-of-domain documents. *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp.210-219). ACM.

Long, M., Wang, J., Ding, G., Cheng, W., Zhang, X., & Wang, W. (2012, April). Dual transfer learning. In *Proceedings of the 2012 SIAM International Conference on Data Mining* (pp. 540-551).

Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge & Data Engineering*, 22(10), 1345-1359.

Luong, M. T., & Manning, C. D. (2015). Stanford neural machine translation systems for spoken language domains. In *Proceedings of the International Workshop on Spoken Language Translation*.

Terminology Translation Accuracy in Phrase-Based versus Neural MT: An Evaluation for the English-Slovene Language Pair

Špela Vintar
University of Ljubljana, Department of Translation Studies
Aškerčeva 2, SI - 1000 Ljubljana
spela.vintar@ff.uni-lj.si

Abstract

For specialised texts, the accuracy and consistency of terminology is of primary importance, yet most Machine Translation systems do not employ explicit strategies to ensure term consistency on the level beyond a single sentence. We present a multifaceted evaluation and comparison of a statistical phrase-based versus neural model of Google's translation system for the English-Slovene language pair, which consists of a document-based automatic evaluation with the BLEU and NIST metrics, an automatic evaluation of term translations using an existing termbase as reference, and a human evaluation of 300 sample sentences per MT model and translation direction. Results indicate that while neural MT regularly outperforms phrase-based MT in the overall scores, the accuracy of term translations is better only for the English-Slovene language pair and not in the Slovene-English translations. In the final part of the paper we discuss typical errors encountered in the different MT outputs.

Keywords: MT evaluation, terminology, neural machine translation, terminology in MT

1. Introduction

Neural Machine Translation (NMT) is quickly becoming mainstream and has been shown to outperform statistical, mainly phrase-based, systems (SMT) across a number of features. Most of the reported evaluations so far (Machacek and Bojar 2014, Bachdanau et al. 2015) rely on automatic metrics and show consistent improvement for almost all tested language pairs. Some authors recently performed more detailed comparisons of statistical vs. neural systems using human evaluators and a more detailed error typology (Bentivogli et al. 2016, Klubička et al. 2017), or focusing on specific properties of the machine translated output such as fluency or reordering (Toral and Sánchez-Cartagena 2017). While these fine-grained evaluations bring additional evidence that NMT represents a giant leap towards more human-like translations, results obtained in some error categories, e.g. lexical or terminology errors, are not as straightforward and do not always support the NMT's claims for supremacy.

In professional translation environments, terminology research takes up to 45% of the total working time spent on translating a text, and according to a recent study by SDL[1] terminology errors amount to over 70% of all errors found in the Quality Assurance (QA) process. Post-editing guidelines developed by organisations such as TAUS[2] or SDL[3] suggest that post-editors should pay particular attention to the consistency of terminology, because nearly all state-of-the-art MT systems still produce translations on a segment-by-segment basis and thus choose terms according to local contexts instead of entire texts.

The aim of this paper is to evaluate the quality of Google Translator (GT) NMT model (Wu et al. 2016) compared to its earlier phrase-based (PBMT) model for the Slovene-English language pair and in the specialised domain of karstology. Google released the NMT model for Slovene-English in October 2017 and to our knowledge no systematic comparison of both models has been performed to date. Apart from the automatic evaluation using metrics we specifically focus on the translations of domain-specific terms, where we describe an experiment combining automatic and manual evaluation of the translation accuracy for karstology terms.

2. Methods and Data

2.1 The Karst Corpus and Termbase

For our evaluations we used a parallel corpus of scientific abstracts and articles pertaining to karstology from two international journals, Acta Carsologica and Acta Geographica Slovenica. Both of these journals publish papers with abstracts in Slovene and English, and the latter translates entire articles either into Slovene or English so that the entire journal is fully bilingual with translations in both directions.

For our experiment we use 20 parallel texts, of which 15 were abstracts and 5 entire articles. The total size of the English part of the corpus is 25,423 tokens and 18,985 tokens for Slovene. All texts were translated twice, first with the PBMT model and then with the NMT model, both provided through the GT API. We translated and evaluated in both directions, English-Slovene and Slovene-English.

It might perhaps seem futile to evaluate a general purpose MT system such as GT on a domain-specific corpus. However, we selected the domain of karstology because it is a relatively well-known field in Slovenia with a large overlap with general language. Over 45% of Slovenian landscape is karst with some of the largest and most visited tourist attractions such as the Postojna or Škocjan Caves. As a consequence, there exist numerous online sources, often bilingual, from which general MT systems such as GT might obtain their training data.

For the evaluation of term translations we rely on the quadrilingual terminology database of karst terms

[1] http://www.sdl.com/download/the-importance-of-terminology-management/71096/

[2] https://www.taus.net/knowledgebase/index.php?title=Category:Post-edit

[3] http://www.sdl.com/download/introduction-to-machine-translation-and-postediting-paradigm-shift/58317/

compiled within the QUIKK project[4]. For the Slovene-English language pair the termbase contains 81 full entries with Slovene and English single- and multiword terms, definitions and other types of information. The termbase is concept-oriented so that it may contain several expressions for the same concept. Thus, for the concept defined as *large flat surface in karst formed by erosion and corrosion* we find the English terms *karst plateau* and *karst plain*, and the Slovene terms *kraška planota*, *kraška uravnava* and *kraški ravnik*.

2.2 Evaluation Methods

For the automatic evaluation of overall performance we use the BLEU (Papineni et al. 2002) and NIST (Doddington 2002) metrics, the former because it is the most widely used and the latter because it has been found to correlate well with human judgements on the document level (Peterson and Przybocki 2010). Since the initial inspection of the translations with the naked eye showed considerable variation in quality, we decided to compute the metrics for each text separately to be able to observe the variation in scores.

Next we approached the evaluation of terms and their translations. For the automatic part of the evaluation we simply identify terms in the original texts using the QUIKK termbase and check whether the aligned translated segment contains the correct equivalent. Because Slovene is a language with rich morphology, both the Slovene terms and the corpus were lemmatised to facilitate matching. Still, the termbase is relatively small and in addition focuses on karst landforms, we decided to complement these results with human evaluation to assess the translations of terms other than those found in the termbase.

For the manual evaluation we first considered using the MQM framework (Lommel et al. 2014), but decided against it because our specific focus is terminology and we would thus be able to use only the error category Mistranslation, which subsumes Terminology. Instead we produced evaluation sets of 300 random term occurrences for both systems and translation directions, which were manually checked by a domain expert. Three categories were used to annotate the term translations found in machine-translated sentences:

- **Correct**, meaning that the translation of the term is the correct equivalent in the selected domain, however regardless of the agreement, case, number or other grammaticality issues,
- **False**, meaning that the word or phrase in the translation is not the appropriate equivalent in karstology. In some cases the MT system used a more generic but still accurate expression; in such cases the domain expert used common sense to decide whether it was correct or false in the given context. For example, the karstology termbase lists *precipitation* and *precipitacija* as equivalents, but the system used *padavine* in Slovene, which is synonymous to *precipitacija* and was confirmed by the domain expert as a possible translation. For multi-word terms, a partially correct translation was considered wrong.
- **Omitted**, meaning that the term from the original sentence was skipped in the translation.

In the following sections we describe the results and discuss the types of errors found.

3. Results

3.1 Automatic Evaluation

The texts deal with different topics within the domain of karst and contain varying ratios of domain-specific terms, which may help explain the high variations in the BLEU and NIST scores obtained (Table 1). For English-Slovene, the average BLEU score of 18.50 for PBMT ranges from 4.55 to 36.26, and the NMT average of 22.49 shows an even higher standard deviation (8.85) with scores from 6.79 to 43.41. Looking at individual BLEU scores, NMT outperforms PBMT for 16 out of 20 texts, and 15 out of 20 according to NIST scores. The latter do not always correlate with BLEU as the highest score of 5.92 was assigned to the NMT translation of article AGS3, which received "only" 28.42 BLEU points.

	English-Slovene				**Slovene-English**			
	PBMT		**NMT**		**PBMT**		**NMT**	
	BLEU	NIST	BLEU	NIST	BLEU	NIST	BLEU	NIST
AC1	26.26	4.56	30.72	4.78	26.1	4.73	31.12	5.03
AC2	7.86	2.36	10.85	2.58	16.95	3.77	15.66	3.70
AC3	16	2.53	15.04	2.48	14.23	3.15	19.77	3.54
AC4	24.84	4.03	34.47	4.69	26.99	4.41	27.65	4.38
AC5	4.55	1.56	6.79	1.51	6.37	1.72	8.83	2.04
AC6	18.3	3.13	20.35	2.93	28.87	3.97	34.1	4.28
AC7	36.26	4.92	43.41	5.09	38.14	5.40	40.93	5.24
AC8	17.76	3.29	22.57	3.77	24.23	4.02	24.13	4.00
AC9	15.06	3.43	31.81	4.30	21.85	4.21	35.75	5.06
AC10	15.01	3.52	18.14	4.12	23.19	4.34	23.32	4.28
AC11	19.6	3.65	22.54	3.78	26.12	4.25	25.97	4.46
AC12	11.76	2.45	11.05	2.19	17.49	3.11	17.63	3.10
AC13	8.04	2.09	11.94	2.47	16.09	3.33	11.4	3.15
AC14	21.41	3.87	29.3	4.28	27.11	4.71	35.92	4.79
AC15	20.96	3.45	24.08	3.85	22.93	4.16	27.25	4.39
AGS1	25.77	5.08	23.89	4.91	22.6	5.28	23.24	5.28
AGS2	21.69	4.47	21.3	4.54	21.71	4.87	24.98	4.99
AGS3	22.02	5.24	28.42	5.92	23.11	4.78	28.11	4.53
AGS4	13.49	3.41	17.21	3.78	19	4.85	23.47	5.08
AGS5	23.28	4.75	25.97	5.13	27.55	5.76	29.38	5.76
Average	**18.50**	**3.59**	**22.49**	**3.85**	**22.53**	**4.24**	**25.43**	**4.35**
St. dev.	7.24	1.02	8.85	1.13	6.41	0.90	7.97	0.88

Table 1: BLEU and NIST scores for the En-Sl and Sl-En language pairs

[4] http://islovar.ff.uni-lj.si/karst

For Slovene-English, the scores are on average slightly higher with 22.53 BLEU for PBMT and 25.43 for NMT, and a moderate improvement in the NIST score from 4.24 to 4.35 respectively. It also seems that the average quality is slightly more consistent with English as the target, as the standard deviation is lower than for En-Sl in all four sets of scores. Again, NMT achieves higher BLEU scores for 16 out of 20 texts.

3.2 Evaluating Term Translations

When we automatically checked for the occurrence of terms from the termbase in the original and the presence of the correct equivalent in the translated segment, the results were inconclusive (Table 2). For the English-Slovene language pair NMT appears to choose the correct equivalent slightly more often than PBMT, while the opposite direction shows a reversed picture with PBMT outperforming NMT by 30 correct translations. It should be noted however that the figures below represent term occurrences and not different terms, thus a large portion of these examples (over 300) were simply occurrences of the terms *karst* (Sl. *kras*) and *cave* (Sl. *jama*) which were for the most part translated correctly by both systems. Of course in many cases these two words occurred within a multi-word term, but if the multi-word term was not recorded in the termbase we could not automatically detect it.

	En-Sl		Sl-En	
	PBMT	NMT	PBMT	NMT
Terms in original	538	538	680	680
Correct terms in translation	420 (78%)	431 (80%)	476 (70%)	446 (65.5%)

Table 2: Terms and equivalents matching the termbase

A detailed insight into the performance of both MT system versions and the types of errors they make can only be gained through human evaluation where we consider the full terminological inventory of the texts. Here, the domain expert was advised to assess not only other multi-word terms but also the translation of proper names referring to relevant places in karst (*Divača karst, Postojna Cave*) which can be especially tricky due to the rich morphology and complex capitalisation rules in Slovene (*Divaški kras, Postojnska jama*). On the other hand, grammatical errors were not to be considered, so that a correct term in the wrong case would still be marked as correct, and the overall fluency or semantic accuracy of the sentence was not part of this evaluation.

	En-Sl				Sl-En			
	PBMT	%	NMT	%	PBMT	%	NMT	%
Correct	184	61.3	211	70.3	201	67	195	65
False	113	37.7	85	28.3	94	31.3	99	33
Omitted	3	1	4	1.3	5	1.7	6	2

Table 3: Human evaluation of term translations

Table 3 lists the results of the human evaluation of term translations in our dataset. The best performance is achieved by NMT in the English-Slovene translation direction where over 70% of the terms were translated correctly, which is a marked improvement from 61% achieved by PBMT. However, the results for the Slovene-English language pair are less conclusive with an insignificant difference between the two system variants and with NMT performing slightly lower than PBMT, which is in line with the results from the automatic evaluation.

3.3 A Glance at Errors

In the English-Slovene PMBT translations, the following types of errors are most common:

- untranslated term or term component (*epigenic aquifer → epigenic vodonosnik, solution runnel → raztopina runnel, hypogenic system → hypogenic sistem, paleokarst → paleokarst*)
- ambiguous term translated with the wrong sense for the domain (*spring* /as in water spring/ → *vzmet* /as in technical domains flexible metal part/, *Mlava Spring → Mlava pomladi* /spring as season of the year/, *solution* /as in water solution/ → *rešitev* /as in solution of a problem, *cave chamber → jamski zbornice* /as in chamber of commerce/)
- errors in translations of terms containing proper names (*Carpathian karst → Karpatih kras, Divača karst → Divača kras*)
- "strange" errors, such as *karst → kra* /which is a non-existent wordform in Slovene/

In the NMT translations we encounter even more examples of translations which are difficult to explain, but on the other hand NMT is creative in coining translations of unknown terms:

- *cave diving → jalovo potapljanje* /jalovo means barren or futile/, *karst processes → krasni procesi* /krasni means splendid/
- *non-paleokarstic rocks → nepaleokarstične kamnine, non-karst areas → nekarska območja, glaciation → glacijacija, aerially exposed → ajerno izpostavljeni* /nepaleokarstični, nekarska, glacijacija, ajerno are all newly coined words in Slovene)

For the Slovene-English language pair, PMBT makes similar types of errors as described before, but fewer ambiguity-related errors:

- untranslated terms (*nepaleokraške kamnine → nepaleokraške rocks, kompetitorskih vrst → kompetitorskih species, pobočja vadijev → vadijev slopes*)
- wrong or non-terminological translation (*brezstropa jama → roofless cave* /instead of denuded cave/, *jamski rov → underground tunnel* /instead of cave passage/, *udornica → hollow, precipice, collapsed, sinkhole* /instead of collapse doline/)
- some confusion with geographical names (*reka Reka → river River, Kras → Karst* /instead of Kras when it refers to the Kras plateau/), although great consistency in the translations of *Divaški kras → Divača karst* or *Škocjanske jame → Škocjan Caves.*

NMT from Slovene into English has other types of problems:

- "strange" translations, possibly due to wrong decomposition of the source term (*vrtač → crop*

rotation /instead of sinkhole/) or simply inexplicable (*zakraselost → naivety, zakrasele planote → plumed plateaus* /instead of karstification and karstified plateaus/, *melioracija → reclamation*)
- great inconsistencies for the term *udornica* (*udornica → collapse, udder, cliff, collision, burrow, groove* /instead of collapse doline/)
- unsuccessful attempts of generating the correct form of proper names (*Senožeški potok → Senožeški brook, Divaški kras → Divaški karst, Divačski karst; Orehovški kras → Orehovsk karst, Orehovska karst, Orehovsky karst*).

It would appear that lexical choice and disambiguation are still areas where NMT systems have significant room for improvement, despite the fact that NMT translations often indeed appear more fluent or natural than PBMT.

4. Discussion and conclusions

It is common wisdom that if we want an MT system to be good at tackling terminology and translating specialised texts, we should train or customize it for the domain of choice. But in many professional translation settings such customization is not easily integrated into the daily workflow, and many freelance translators work in multiple domains. There have been interesting attempts to facilitate such customization and help users "inject" bilingual terminologies into an existing MT system used in a computer-assisted translation (CAT) environment (Arčan et al. 2014). Still, in many cases the "general purpose" MT system is used to translate specialised content without customization.

According to itself, GT serves over 500 million users monthly and translates over 140 billion words per day, which is more than the entire language industry produces in a year[5]. Given these volumes it becomes clear that a considerable portion of this input must be specialised. NMT has clearly improved the fluency of translated output and will likely continue to amaze with methods for the handling of unknown words, it seems however that the accuracy and consistency of terminology still leave room for improvement.

Our evaluation of GT's phrase-based and neural models for the English-Slovene language pair in both translation directions was primarily aimed at testing whether NMT performs better on domain-specific texts, whereby a focused automatic and human evaluation was performed for the accuracy of term translations. A general evaluation with metrics indicates that NMT indeed produces better quality translations in both directions, however for terminology such an improvement was observed only for the English-Slovene translations and not vice versa.

5. Bibliographical References

Arčan, M., Turchi, M., Tonelli, S., & Buitelaar, P. (2014). Enhancing statistical machine translation with bilingual terminology in a CAT environment. In *Proceedings of the 11th Biennial Conference of the Association for Machine Translation in the Americas (AMTA 2014)* (pp. 54-68).

Bahdanau, D., Cho, K. and Y. Bengio (2015). Neural machine translation by jointly learning to align and translate. In Proc. of ICLR, San Diego, US-CA.

Doddington, G. (2002). Automatic evaluation of machine translation quality using N-gram CoOccurrence statistics, in Proc. of Second International Conference on Human Language Technology (HLT), San Diego, March 2002, pp. 138-145.

Klubička, F., Toral, A., & Sánchez-Cartagena, V. M. (2017). Fine-grained human evaluation of neural versus phrase-based machine translation. The Prague Bulletin of Mathematical Linguistics, 108(1), 121-132.

Lommel, A., Uszkoreit, H., & Burchardt, A. (2014). Multidimensional Quality Metrics (MQM). *Tradumàtica*, (12), 0455-463.

Machacek, M., and Bojar, O. (2014, June). Results of the WMT14 Metrics Shared Task. In *WMT@ ACL* (pp. 293-301).

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). BLEU: a method for automatic evaluation of machine translation. In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, ACL '02, pages 311–318.

Peterson, K. and M. Przybocki, NIST 2010 Metrics for Machine Translation Evaluation (MetricsMaTr10) Official Release of Results, http://www.itl.nist.gov/iad/mig/tests/metricsmatr/2010/results

Toral, A., & Sánchez-Cartagena, V. M. (2017). A Multifaceted Evaluation of Neural versus Phrase-Based Machine Translation for 9 Language Directions. arXiv preprint arXiv:1701.02901.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., ... and Klingner, J. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144.

[5] https://translate.google.com/intl/en/about/

A Semi-supervised Learning Approach for Person Name Recognition in Tibetan

Zhijuan Wang[1,2], **Fuxian Li**[3]
Minzu University of China[1], Alibaba[3]
National Language Resource Monitoring and Research Minority Language Center[2]
No.27, South Street of Zhongguancun, Haidian District, Beijing, China[1,2]
No.8, Haidian Street, Haidian District, Beijing, China[3]
{wangzj.muc, fushine.lee}@gmail.com

Abstract

Massive labelled data is important for Named Entity Recognition(NER). For Low Resource Languages(LRL), massive labelled data means more labor, more time and more cost. A semi-supervised learning (SSL) that need fewer labelled data is proposed to recognize person name in Tibetan texts. Based on Conditional Random Fields (CRFs) and Radial Basis Function (RBF), this method use 5-element feature matrix to propagate information from few labeled data to massive unlabelled data. Experiments demonstrate that its F-measure can achieve 84% using only 100 documents as seeds, whereas about 800 labeled documents are required for a supervised learning based on pure CRFs.

Keywords: Low Resource Languages, Person Name Recognition, Semi-supervised Learning

1. Introduction

Named Entity Recognition (NER), whose main task is to recognize the names of persons, locations and organizations from texts in different languages texts, is an important task for information extraction (IE), Information Retrieval (IR)Information Retial. As mentioned in (David and Satoshi, 2007) (Chung et al., 2003) (Popov et al., 2004) (Benajiba et al., 2007) (Seker and Eryigit, 2012), NER research had covered many languages such as English, German, Spanish, Chinese, Japanese, Korean, Russian, Arabic, Turkish and so on.

Machine learning approaches such as CRFs, HMM often be used in NER. According to the size of labelled data, machine learning approaches can be divided into supervised learning(need massive labelled data), semi-supervised learning(need a small amount of labelled data) and unsupervised learning(no labelled data). Supervised learning has the better performance in NER. In order to make up for the deficiency of the labelled data,some semi-supervised methods (Nadeau et al., 2007) and unsupervised methods(Michael et al., 2005) are used.

Tibetan is a low resource language which is a cluster of Sino-Tibetan languages and spoken primarily by Tibetan peoples, who live across a wide area of eastern Central Asia. There are some research focused on Tibetan NER, especially on person name recognition. These methods are all based on rules or supervised learning approaches. Very few efforts have been made to develop semi-supervised learning or unsupervised learning for Tibetan NER.

A Semi-supervised Learning Approach is proposed to recognize Person Name in Tibetan.The remainder of this paper is structured as follows: Section 2 includes background information about the features of Tibetan person name and recent work of Tibetan person name recognition. Section 3 illustrates the methodology of the proposed algorithm. The data used in experiment and the evaluation results are reported and discussed in section 4. Finally, we present the conclusion and future work.

2. Background

There is little introduction about Tibetan person names in English. So, we give a brief introduction of Tibetan person name firstly.

2.1. Introduction of person name in Tibetan

Tibetan is an alphabetic writing language, which has 30 consonants and four vowel signs. Its smallest grammar unit is syllable. ”་” is the mark of syllable. One or more alphabets compose a syllable and one or more syllables can compose a word. Fig.1 is an example of Tibetan sentence with named entities. We can see that there is no white space between Tibetan words and there is no obvious feature such as capitalization of first letter to identify the person name in Tibetan.

Person names in Tibetan are complex. There are four kinds person name in Tibetan text.

(1) Tibetan first name

Most Tibetan people' name only have first names, which length often range from two syllables to five syllables. For example, ”དཔལ་བཟང་ ” (Passang), ”པད་མ་འཚོ” (Pematso), ”བློ་བཟང་བྱམས་པ་ ” (Lobsang Champa). First names of Tibetan people often come from Buddhism or other good wish. Therefore, some first names often be used, which are called high-frequency syllables of Tibetan people'name. For example, ”སྒྲོལ་མ་ ” (Dolma), ”བཀྲ་ཤི་” (Tashi).

(2) Chinese surname name + Chines first name

There are a large number of Transliteration of Chinese person names in the Tibetan text, which may come from Tibetan people(some Tibetan people use Chinese person name) or Chinese people. For example, ”ལས་ཅ་ཁྲིང་” (Li Ka-shing).

(3)Chinese surname name + Tibetan first name

Some Tibetan people's names not only have Tibetan first name, but also Chinese surname. For example, ”ལི་སྒྲོན་མ་ ” (Li Dolma) is Tibetan people name. ”ལི་” (Li) is Chinese Surname, ”སྒྲོན་མ་་” (Dolma) is Tibetan first name.

(4) Tibetan surname name + Tibetan first name

Generally, only Tibetan nobility have Tibetan surname. For example, ”ང་ཕོད་ངག་དབང་འཇིགས་མེད་” (Ngapoi Ngawang Jigme) is Tibetan people name. ”ང་ཕོད་ངག་” (Ngapoi) is Tibetan Surname, ”དབང་འཇིགས་མེད་” (Ngawang Jigme) is Tibetan first name.

Since the latter two kinds of person names share a small proportion in the Tibetan text, we focus on how to identify the first two kinds person names in the Tibetan text.

2.2. Related Work of NER in Tibetan

The research of Tibetan NER are focused on two approaches.

Rules: (Yu and Jiang, 2010) utilized a rule-based model on case-auxiliary words, lexicon and boundary information list to recognize Tibetan named entity; also, (Sun et al., 2010) used multi-features such as internal features, contextual features and boundary features for recognition task.

Supervised learning: (Jin et al., 2010) uses rules and Hidden Markov model(HMM) to Tibetan NER.(Jia et al., 2014) combines Maximum Entropy (MaxEnt) and Conditional Random Fields (CRFs) to identify Tibetan person names. (Hua et al., 2015) proposed a Perception Training Model based on Tibetan syllable features to identify Tibetan NER. (Kang et al., 2015) and (zhu et al., 2005) used CRFs to recognize Tibetan person names.

The above approaches based on rules and supervised learning require Tibetan linguists construct rules or native speakers to annotate a lot of training data. (Jia et al., 2014)'s approach based on CRFs and MaxEnt needs 3.5 MB training data; (Hua et al., 2015)'s training data contain 15,000 sentences; (Kang et al., 2015)'s CRF-based method takes 40,000 words as training data.

In this paper, we propose a semi-supervised learning approach to recognize the person name in Tibetan to reduce the human labor and budgets.

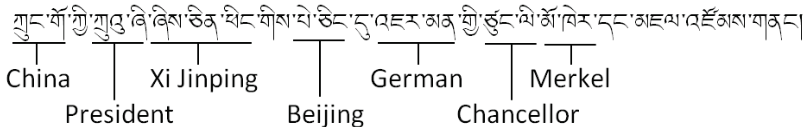
ཀྲུང་གོ་ཀྱི་ཀྲུའུ་ཞི་ཤི་ཅིན་ཕིང་གིས་པེ་ཅིང་དུ་འཇར་མན་གྱི་ཚུང་ལི་མོ་ཁེར་དང་མཇལ་འཛོམས་གནང་།
China
President
Xi Jinping
Beijing
German
Chancellor
Merkel

Figure 1: An example of named entities in Tibetan

3. Methodology

There are some often-used semi-supervised learning methods, including: EM with generative mixture models, self-training, co-training, transductive support vector machines, and graph-based methods(Zhu, 2005). And some supervised learning algorithm can also be used in semi-supervised learning algorithm ((Jiao et al., 2006),(Mann and McCallum, 2010),(Liu et al., 2011)). Here, we propose a semi-supervised learning based on Conditional Random Fields (CRFs) and Radial Basis Function (RBF) to recognize person name in Tibetan.

There are two reasons that we adopt CRFs to realize our semi-supervised learning approach. Firstly, CRFs(Lafferty et al., 2001) are a type of discriminative undirected probabilistic graphical model. Because the model is conditional, dependencies among the input variables do not need to be explicitly represented, affording the use of rich, global features of the input (Sutton and McCallum, 2006), CRFs are often applied in named entity recognition (McCallum and Li, 2003) (Settles, 2004). Secondly, we can train CRFs model based on Tibetan syllable, which need not word segmentation of Tibetan.

3.1. Methods

We assume that there are l labelled points $(x_1, y_1), ..., (x_l, y_l)$ and u unlabelled points $x_{l+1}, ..., x_{l+u}$; typically $l << u$. Using L and U as labelled points set and unlabelled points set separately. We suppose the labels are binary.

$$y_L = \begin{cases} 1 & \text{Person names} \\ 0 & \text{Otherwise} \end{cases} \tag{1}$$

The semi-supervised learning algorithm based on CRFs and RBF is shown in Algorithm 1.

Algorithm 1:

Given:

L: a small set of labelled training data. There are m annotated person names $PER_j, j \in m$.

U: a lot of unlabelled training data.

Training Model

Step1: train a CRFs model M_L based on L.

Step2: use M_L to classify unlabelled data U and get n labelled person names $MPER_i, i \in n$.

Step3: extract 5-element feature matrixes of PER_j and $MPER_i$, and calculate their similarities based on RBF. Afterwards, select the biggest K similarity values for every annotated entity $MPER_i$ using k-Nearest Neighbors (KNN) algorithm. Then, calculate the mean $Sim(MPER)_i$ of the K similarity values as the similarity of $MPER_i$ to PER_j.

Step4: extract the data which have the most similarity to PER_j and add this data into seeds set L. Meanwhile, remove them from unlabeled data U.

Step5: If the algorithm is converged or the number of loops reached the max iteration, then end this algorithm, else go to step 1.

3.2. Seeds selection

For semi-supervised learning, a small amount of labelled data, which can be called gold seeds, are very important. For person names recognition in Tibetan, in order to ensure the precision and efficiency of model, the gold seeds should cover the important features, such as:

(1) Tibetan person names, transliteration names from Chinese and other foreign countries.

(2) Tibetan person named with titles and case-auxiliary words, Tibetan person named without titles and case-auxiliary words.

(3) Some words which can be used as person name as well as ordinary nouns. For example, ”ཉི་མ་” (Nyima) can be a Tibetan people name or ordinary nouns ”Moon”.
The golden seeds in our experiment based on this requirement.

3.3. Feature selection for person name recognition in Tibetan

Seeds propagation is crucial for semi-supervised learning. We use feature matrix of person name in Tibetan to realize the seed propagation. Therefore, we will discuss the feature selection of the person name in Tibetan firstly.
Here, two kinds features are used: initial feature and context feature.

(1) Initial features

The initial features are often used in NER in many languages. For example, capitalization and family names are used in the NER of English and Chinese. Tibetan people names and transliteration of Chines person have different initial features. For Tibetan person names, high-frequency syllables can be initial features. Meanwhile, Chinese Surname can be initial feature for transliteration of Chinese person names.
Using 10,460 (41,755 syllables) Tibetan person names, we select some Tibetan person names that their frequencies are exceed 1% as high-frequency syllables. The top 5 examples are shown in Tab.1.

Tibetan	English
བཀྲ་ཤིས་	Tashi
ཚེ་རིང་	Tsering
བསྟན་འཛིན་	Tenzin
སྒྲོལ་མ་	Dolma
ཉི་མ་	Nyima

Table 1: High-frequency syllables of person names of Tibetan people.

For 504 Hundred Family Surnames, 444 are single-character surnames and 60 are double-character surnames. Because some Chinese Family Surnames have same pronunciation, the 504 Hundred Family Surnames can be translated in 291 tibetan syllables. The example is shown Tab.2.

Tibetan	English
ཝང་	Wang
ལི་	Li
ཀྲང་	Zhang
ཡན་	Yan
ཝུའུ་	Wu
ཀུང་	Gong

Table 2: Some Chinese surname in Tibetan

(2) Context features: Two features, case-auxiliary word and title, are used in this paper as context features.

Case-auxiliary word is one of the most important components for Tibetan. Among eight kinds of Case-auxiliary, two of them are used as features to recognize Tibetan person name, do-case-auxiliary words and belong-case-auxiliary words, because they are often appeared after or before person name in Tibetan.
do-case-auxiliary words:
གིས་, ཀྱིས་, གྱིས་, འིས་, ཡིས་.
belong-case-auxiliary words:
གི་, ཀྱི་, གྱི་, འི་, ཡི་.

Tile is an important feature for NER task. For person names recognition in Tibetan, two kinds title are used. The first is traditional title such as president, chairman. The other is special titles that are unique for Tibetan. The example is shown Tab.3.
The position of **title** in Tibetan is different from the position in other languages(English, for example) since it can be inserted before or after a person name.
Therefore, in this paper, we use five features to extract Tibetan person name: high-frequency syllables (include high-frequency syllables of Tibetan people’s names and Chinese family names); left/right title (the title appear before or after person names); left/right case-auxiliary

Tibetan	English
ཙུང་ཐུང་	President
ཀྲུའུ་ཞི་	Chairman
ཙུང་ལི་	Minister
པན་ཆེན་བླ་མ་	Panchen Lama
རིན་པོ་ཆེ་	Rinpoche
སྤྲུལ་སྐུ་	Tulku

Table 3: The example of Tibetan Title

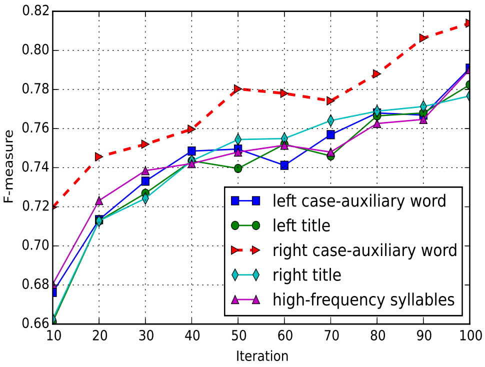

Figure 2: The influence of features to recognize Tibetan person names

word (the case-auxiliary word appear before or after person names).

The influences of five features on person name recognition in Tibetan is shown in Fig.2. We can see that right case-auxiliary word has the biggest influence comparing with the influences of other features.

3.4. Seeds selection

For semi-supervised learning, the annotated data, which can be called gold seeds, are very important. For Tibetan person names recognition task, in order to ensure the precision and efficiency of model, the gold seeds should cover the important features, such as:

(1) Tibetan person names, transliteration names from Chinese and other foreign countries.

(2) Tibetan person named with titles and case-auxiliary words, Tibetan person named without titles and case-auxiliary words.

(3) Some words which can be used as person name as well as ordinary nouns. The golden seeds in our experiment based on this requirement.

3.5. Seed propagation

Using M_L gotten by CRFs, the unlabeled data U can be annotated. Take annotated entities PER_j and new labeled entities $MPER_i$ as nodes V to construct graph $G = (V, E)$. E are edges. We assume an $i * j$ symmetric matrix W on the edges of the graph is given. Then, the similarities of $MPER_i$ andPER_j, w_{ij}, can be calculated by RBF (Zhu et al., 2003)in Formula 2.

$$w_{ij} = exp(-\sum_{d=1}^{5} \frac{\beta_d \cdot (x_{id} - x_{jd})^2}{\sigma^2}) \qquad (2)$$

Where x_{jd} and x_{id} is the $d - th$ feature of PER_j and $MPER_i$ respectively.

As shown in Fig.2, the influences of different features on Tibetan person name recognition are different. So, we give 5 features with different weights. β_d is feature weight.

Then, we can calculated $Sim(MPER)_i$ using formula (2) and k-NN graph (k=5)(Zhou et al., 2004) in Formula (3).

$$Sim(MPER_i) = \frac{1}{K} \sum_{x_j} w_{ij}, x_j \in KNN(x_i) \qquad (3)$$

4. Evaluation

We used 1100 documents from websites (tibet.people.com.cn, from 2015-2017) as experiment data.

In our experiment, some documents are selected as gold seeds according to the principle of seeds selection. 100 documents are selected as test data. The reminder documents are unlabelled data.

Labelled Data(Documents)	F-measure%
100	45.23
200	59.78
300	66.21
400	73.75
500	75.93
600	78.58
700	82.77
800	84.12
900	86.73
1000	90.31

Table 4: F-measure of Tibetan person name based on CRFs.

4.1. The baseline

We train a baseline model based on (CRF++-0.58) using 1000 annotated documents.Its precision, recall and F-measure are shown in Table 2.

(Kang et al., 2015)'s F-measure is 94.31% of, which use CRFs and some features to recognize Tibetan person names. Our baseline use CRFs and less training data. So, the F-measure (90.31%) of baseline is acceptable.

4.2. The influence of multi-feature

In the process of seed propagation, 5 features were given with different weights. The influence of features on F-measure based on 100 gold seeds are shown in Fig.3. We can see that the performance of multi-feature with different weights is better at least 3% than the performances of other single features.

4.3. The influence of seeds

Fig.4 shows the relationship of F-measure and iterations of 50, 60, 70, 80, 90, 100 seeds. As the number of seeds increases, the F-measure increases. The highest F-measure of Tibetan person names recognition can reach about 84% when 90 or 100 seeds iterate 100 times.

For semi-supervised learning, less seeds means less annotated data and low cost of money and time. Therefore, we can use about 100 documents as golden seeds to extract person name in Tibetan.

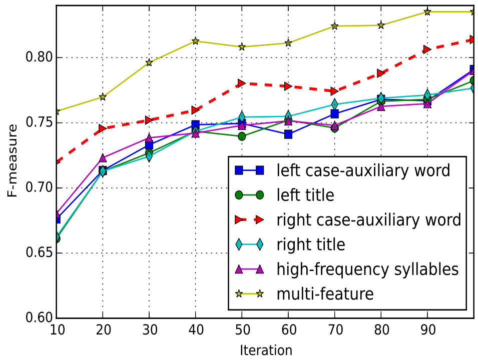

Figure 3: the influence of feature on F-measure

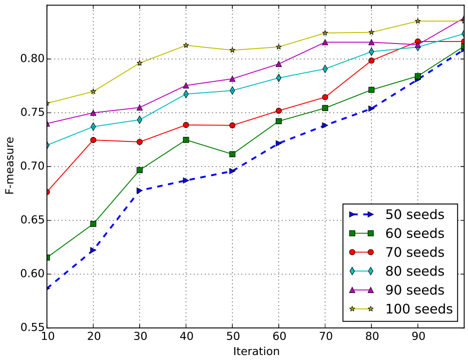

Figure 4: the influence of seeds on F-measure

4.4. The comparation of semi-supervised learning and supervised learning

Here, we will compare the semi-supervised leaning based on CRFs and RBF to supervised learning method based on CRFs.

As shown in Table 3, the semi-supervised approach achieves much better results than supervised approach when the same amount labelled documents are used. Using only 100 annotated documents, the F-measure of semi-supervised

Labelled data	F-measure
supervised (100 documents)	24.24
supervised (800 documents)	84.12
semi-supervised (100 documents)	83.82

Table 5: The comparison of semi-supervised learning and supervised learning.

learning approach can reach 84%, whereas about more than 800 labelled documents are required for a supervised learning approach based on pure CRFs.

5. Conclusion

For low resource language such as Tibetan, the methods of person name recognition based on supervised learning need a lot of annotated data, which means more human labor, higher budget, and more time. We propose a semi-supervised learning (SSL) approach based on Conditional Random Fields (CRFs) and Radial Basis Function (RBF) to recognize Tibetan person names. And Five feature (high-frequency syllables, left/right title and left/right case-auxiliary words) are used to propagate information from labelled documents to unlabelled data. Experiments demonstrate that this method can recognize person name in Tibetan at low cost with an acceptable performance.

In the future, we will try to construct a common system to extract person name in other low resource language, which based on CRFs, RBF and feature matrix at three level (word level, context level and sentence level). Moreover, we will try to improve the efficiency of propagation.

6. Acknowledgements

This work was supported by the China National Natural Science Foundation No. (61331013).

7. References

Yassine Benajiba, Paolo Rosso, and José Miguel Benedíruiz. 2007. Anersys: An arabic named entity recognition system based on maximum entropy. In *Computational Linguistics and Intelligent Text Processing*, pages 143–153. Springer.

Euisok Chung, Yi-Gyu Hwang, and Myung-Gil Jang. 2003. Korean named entity recognition using hmm and cotraining model. In *Proceedings of the sixth international workshop on Information retrieval with Asian languages-Volume 11*, pages 161–167. Association for Computational Linguistics.

Nadeau David and Sekine Satoshi. 2007. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1):3–26.

Quecairang Hua, Wenbin Jiang, Haixing Zhao, and Qun Liu. 2015. Tibetan name entity recognition with perceptron model. *Computer Engineering and Application*, 50(15):172–176.

Yangji Jia, Yachao Li, Chengqing Zong, and Hongzhi Yu. 2014. A hybrid approach to tibetan person name identification by maximum entropy model and conditional random fields. *Journal of Chinese Information Processing*, 28(1):107–112.

Feng Jiao, Shaojun Wang, Chi-Hoon Lee, Russell Greiner, and Dale Schuurmans. 2006. Semi-supervised conditional random fields for improved sequence segmentation and labeling. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 209–216. Association for Computational Linguistics.

Ming Jin, Huanhuan Yang, and Guangrong Shan. 2010. The studies of named entity recognition for tibetan. *Journal of Northwest University for Nationalities(Natural Science)*, 31(3):49–52.

Caijun Kang, Cunjun Long, and Di Jiang. 2015. Tibetan name recognition research based on crf. *Computer Engineering and Application*, 51(3):109–111.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Xiaohua Liu, Shaodian Zhang, Furu Wei, and Ming Zhou. 2011. Recognizing named entities in tweets. In *Proceedings of the 49th Annual*

Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, pages 359–367. Association for Computational Linguistics.

Gideon S Mann and Andrew McCallum. 2010. Generalized expectation criteria for semi-supervised learning with weakly labeled data. *The Journal of Machine Learning Research*, 11:955–984.

Andrew McCallum and Wei Li. 2003. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 188–191. Association for Computational Linguistics.

Borislav Popov, Angel Kirilov, Diana Maynard, and Dimitar Manov. 2004. Creation of reusable components and language resources for named entity recognition in russian. In *LREC*.

Gökhan Akin Seker and Gülsen Eryigit. 2012. Initial explorations on using crfs for turkish named entity recognition. In *COLING*, pages 2459–2474.

Burr Settles. 2004. Biomedical named entity recognition using conditional random fields and rich feature sets. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications*, pages 104–107. Association for Computational Linguistics.

Yuan Sun, Xiaodong Yan, Xiaobing Zhao, and Guosheng Yang. 2010. Reseach on automatic recognition of tibetan personal names based on multi-features. In *2010 international conference of natural language processing and knowledge engineering*.

Charles Sutton and Andrew McCallum. 2006. An introduction to conditional random fields for relational learning. *Introduction to statistical relational learning*, pages 93–128.

Hongzhi Yu and Ning Jiang, Tao anf Ma. 2010. Named entity recognition for tibetan texts using case-auxiliary grammars. In *Proceedings of the International MultiConference of Engineers and Computer Scientists*, volume 1.

Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. 2004. Learning with local and global consistency. *Advances in neural information processing systems*, 16(16):321–328.

Xiaojin Zhu, Zoubin Ghahramani, John Lafferty, et al. 2003. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, volume 3, pages 912–919.

Xiaojin Zhu. 2005. Semi-supervised learning literature survey.

OrenEtzioni Michael Cafarella and etc. 2005. Unsupervised named-entity extraction from the Web: An experimental study. *Artificial Intelligence*, 175(1):91–134.

Nadeau, David. 2007. Semi-supervised named entity recognition: learning to recognize 100 entity types with little supervision. University of Ottawa.

Zhu Jie,Li Tianrui,Liu Shengjiu. 2016. Research on Tibetan name recognition technology under CRF *Journal of Nanjing University(Natural Sciences)* , 52(2):289–299.