Machine Learning Approach to Bilingual Terminology Alignment: Reimplementation and Adaptation

Andraž Repar[1,3], **Matej Martinc**[1,2], **Senja Pollak**[2]
[1] Jožef Stefan Postgraduate School, Ljubljana, Slovenia
[2] Jožef Stefan Institute, Ljubljana, Slovenia
[3] Iolar d.o.o., Ljubljana, Slovenia
andraz.repar@iolar.com,{matej.martinc, senja.pollak}@ijs.si

Abstract

In this paper, we reproduce some of the experiments related to bilingual terminology alignment described by Aker et al. (2013). They treat bilingual term alignment as a binary classification problem and train a SVM classifier on various dictionary and cognate-based features. Despite closely following the original paper with only minor deviations - in areas where the original description is not clear enough - we obtained significantly worse results than the authors of the original paper. In the second part of the paper, we try to analyze the reasons for the discrepancy and offer some methods to improve the results. After improvements we manage to achieve a precision of almost 91% and recall of almost 52% which is closer to the results published in the original paper. Finally, we also performed manual evaluation where we achieved results similar to the original paper. To help with any future reimplementation efforts of our experiments, we also publish our code online.

Keywords: term alignment, machine learning, SVM, cognates, word alignment dictionary

1. Introduction

As part of a larger body of work related to bilingual terminology extraction for the needs of the translation industry, we were interested in implementing a machine learning approach to bilingual terminology alignment. The primary purpose of bilingual terminology alignment is to build a term bank - i.e. a list of terms in one language along with their equivalents in the other language. With regard to the input text, we can distinguish between alignment on the basis of a parallel corpus and alignment on the basis of a comparable corpus. For the translation industry, bilingual terminology extraction from parallel corpora is extremely relevant due to the large amounts of sentence-aligned parallel corpora available in the form of translation memories (in the TMX file format). Foo (2012) makes a distinction between two basic approaches: *Extract-align* where we first extract monolingual candidate terms from both sides of the corpus and then align the terms, such as in Vintar (2010), and *align-extract* where we first align single and multi-word units in parallel sentences and then extract the relevant terminology from a list of candidate term pairs, such as in Macken et al. (2013).

However, considerable efforts have also been invested into researching terminology alignment from comparable corpora Daille and Morin (2005) state that there are multiple reasons why one would opt to extract terminology from comparable and not parallel corpora with the most important being that it is often difficult to obtain parallel corpora not involving English. One of the approaches to term alignment on the basis of comparable corpora involves cognates - words that look similar in different languages (e.g. "democracy" in English and "demokracija" in Slovenian), for example Mann and Yarowsky (2001) describe a method that uses cognates to generate bilingual lexicons between languages from different language families.

In this paper, we aim to reproduce the experiments from the paper "Extracting bilingual terminologies from comparable corpora" by Aker et al. (2013) who propose an original approach to bilingual term alignment utilizing machine learning techniques. They treat aligning terms in two languages as a binary classification problem and employ an SVM binary classifier (Joachims, 2002) and training data terms taken from the EUROVOC thesaurus (Steinberger et al., 2002). They construct two types of features: dictionary-based (using word alignment dictionaries created with Giza++ (Och and Ney, 2000; Och and Ney, 2003) and cognate-based (effectively utilizing the similarity of terms across languages).

Despite the problem of bilingual term alignment lending itself well to the binary classification task, there have only been relatively few approaches utilizing machine learning. For example, similar to Aker et al. (2013), Baldwin and Tanaka (2004) generate corpus-based, dictionary-based and translation-based features and train a SVM classifier which returns a continuous value between -1 and 1 which in turn is then used to rank the translation candidates. Note that they only focus on multi-word noun phrases (noun + noun). A similar approach, again focusing on noun phrases, is also described by Cao and Li (2002). Finally, Nassirudin and Purwarianti (2015) also reimplement the approach by Aker et al. (2013) for the Indonesian-Japanese language and further expand it with statistical features (i.e. context heterogeneity similarity). In the best scenario, their accuracy, precision and recall all exceed 90% but the results are not directly comparable since Nassirudin and Purwarianti (2015) use 10-fold cross-validation while Aker et al. (2013) use a held-out test set.

This paper is organized as follows: Section 1 contains the introduction, Section 2 describes the approach by Aker et al. (2013), Section 3 contains our reimplementation efforts, Section 4 describes the approach to improve the reimplementation results, Section 5 contains the results of manual

evaluation and Section 6 contains the conclusions. We also publish our code online for enabling future replicability [1].

2. Description of the original approach

The original approach designed by Aker et al. (2013) was developed to align terminology from comparable (or parallel) corpora using machine-learning techniques. They use terms from the EUROVOC thesaurus and train an SVM binary classifier (Joachims, 2002) (with a linear kernel and the trade-off between training error and margin parameter c = 10). The task of bilingual alignment is treated as binary classification - each term from the source language *S* is paired with with each term from the target language *T*. They then extract features (dictionary and cognate-based) to be used by the classifier. They run their experiments on the 21 official EU languages covered by EUROVOC with English always being the source language (20 language pairs altogether). They evaluate the performance on a held-out term pair list from EUROVOC using recall, precision and F-measure for all 20 languages. Next, they propose an experimental setting for a simulation of a real-world scenario where they collect English-German comparable corpora of two domains (IT, automotive) from Wikipedia, perform monolingual term extraction (based on Pinnis et al. (2012)), followed by the bilingual alignment procedure described above and manually evaluate the results (using two evaluators). They report excellent performance on the held-out term list with many language pairs reaching 100% precision and the lowest recall being 65%. For Slovene, the target language of our interest, the results were 100% precision and 66% recall. The results of the manual evaluation phase were also good, with two evaluators agreeing that at least 81% of the extracted term pairs in the IT domain and at least 60% of the extracted term pairs in the automotive domain can be considered exact translations.

2.1. Features

Aker et al. (2013) use two types of features that express correspondences between the words (composing a term) in the target and source language (for a detailed description see Table 1:

- 7 dictionary-based (using Giza++) features[2] which take advantage of dictionaries created from large parallel corpora of which 6 are direction-dependent (source-to-target or target-to-source) and 1 direction-independent - resulting in altogether 13 features, and
- 5 cognate-based (on the basis of Gaizauskas et al. (2012)) which utilize string-based word similarity between languages.

To capture words with morphological differences, they do not perform direct string matching but utilize Levenshtein Distance. Two words were considered equal if the Levenshtein Distance (Levenshtein, 1966) was equal or higher than 0.95.
Additional features are also constructed by:

- Using language pair specific transliteration rules to create additional cognate-based features. The purpose of this task was to try to match the cognate terms while taking into account the differences in writing systems between two languages. Transliteration rules were created for both directions (source-to-target and target-to-source) separately and cognate-based features were constructed for both directions - resulting in additional 10 cognate-based features with transliteration rules.
- Combining the dictionary and cognate-based features in a set of combined features where the term pair alignment is correct if either the dictionary or the cognate-based method returns a positive result. This process resulted in additional 10 combined features[3].

At the end of the feature construction phase, there were 38 features: 13 dictionary-based, 5-cognate-based, 10 cognate-based features with transliteration rules and 10 combined features.

2.2. Data sources and experiments

Using Giza++, Aker et al. (2013) create source-to-target and target-to-source word alignment dictionaries based on the DGT translation memory (Steinberger et al., 2002). The resulting dictionary entries consist of the source word *s*, its translation *t* and the number indicating the probability that *t* is an actual translation of *s*. To improve the performance of the dictionary-based features, the following entries were removed from the dictionaries:

- entries where probability is lower then 0.05
- entries where the source word was less than 4 characters and the target word more than 5 characters long and vice versa.

The next step is the creation of term pairs from the EUROVOC thesaurus, which at the time consisted of 6,797 terms. Each non-English language was paired with English. The test set consisted of 600 positive (correct) term pairs—taken randomly out of the total 6,797 EUROVOC term pairs—and around 1.3 million negative pairs which were created by pairing a source term with 200 distinct random terms. Aker et al. (2013) argue that this was done to simulate real-world conditions where the classifier would be faced with a larger number of negative pairs and a comparably small number of positive ones. The 600 positive term pairs were further divided into 200 pairs where both (i.e. source and target) terms were single words, 200 pairs with a single word only on one side and 200 pairs with multiple-word terms on both sides. The remaining positive term pairs (approximately 6,200) were used as training data along with additional 6,200 negative pairs. These were constructed by taking the source side terms and pairing

[1] http://source.ijs.si/mmartinc/4real2018

[2] For languages like German, with extensive usage of compounding, additional rules are applied.

[3] For combined features, a word is considered as covered if it can be found in the corresponding set of Giza++ translations or if one of the cognate-based measures (Longest Common Subsequence, Longest Common Substring, Levensthein Distance, Needleman-Wunsch Distance, Dice) is 0.70 or higher (set experimentally by Aker et al. (2013))

each source term with one target term (other than the correct one). Using this approach, Aker et al. (2013) achieve excellent results with results for Slovenian reaching 100% precision and 66% recall.

3. Reimplementation of the approach

As part of a larger body of work on bilingual terminology extraction, we find machine learning approaches interesting because they allow continuous improvement of the output either by fine-tuning or customizing the training set to the output requirements. For this purpose, the approach by Aker et al. (2013) represents a fine starting point for machine-learning-based bilingual term alignment.
The first step in our approach was to reimplement the algorithm described by Aker et al. (2013). The initial premise is the same: given two lists of terms from a similar domain in two different languages, we would like to align the terms in the two lists to get one bilingual glossary to be used in a variety of settings (computer-assisted translation, machine translation, ontology creation etc.). We followed the approach described above faithfully except in the following cases:

- We are focusing only on the English-Slovenian language pair.
- We use newer datasets. The Eurovoc thesaurus currently contains 7083 terms. Similarly, the DGT translation memory contains additional content not yet present in 2013.
- Because our languages (English, Slovenian) don't have compounds, we are not utilizing the approach to compounding described by Aker et al. (2013) for German and some other languages.
- Since no particular cleaning of training data (e.g., manual removal of specific entries) is described in the paper for Slovene, we do not perform any.

We don't think these differences are significant and the experiments should yield similar results.

3.1. Problems with reimplementation

While the general approach is clearly laid out in the article, there are several spots where further clarification would be welcome:

- There is no information about the Giza++ settings or whether the input corpora have been lemmatized. In order to improve term matching, we experimented with and without lemmatization of the Giza++ input corpora.
- There is no information about the specific character mappings rules other than a general principle of one character in the source being mapped to one or more character in the target. Since the authors cover 20 languages, it is understandable that the they cannot include the actual mapping rules in the article. Therefore, we have created our own mapping rules for English-Slovenian according to the instructions in the original paper:
  - Mapping the English term to the Slovenian writing system (the character before the colon is replaced by the sequence of characters after the colon): *x:ks, y:j, w:v, q:k*
  - Mapping the Slovenian term to the English writing system: *č:ch, š:sh, ž:zh*
- We believe that the formula for the Needleman-Wunsch distance in the paper is wrong: instead of $\frac{LCST}{min[len(source)+len(target)]}$ it should be $\frac{LCST}{min[len(source),len(target)]}$ as in Nassirudin and Purwarianti (2015).

We contacted the original authors of the paper and did receive some answers confirming our assumptions (e.g. regarding mapping terms to the different writing systems and that the test set data was selected individually for each language pair), but several other issues remained unaddressed (in particular, what was the exact train and test data selection strategy for the EN-SL language pair). Further inquiries proved unsuccessful due to time constraints on the part of the original authors. We think one of the reasons the lack of clarity of the original paper is its scope: they deal with more than 20 language pairs and it would be impossible to specific information regarding all of them. However, the fact that they deal with all Eurovoc language pairs is also one of the strengths of the original paper.

3.2. Results

The evaluation on the test set of 1,416,600 English-Slovene term pairs shows that compared to the results reported by Aker et al. (2013) (see line 1 in Table 3), our results are significantly worse. Despite all our efforts to follow the original approach, we were unable to match the results achieved in the original paper when running the algorithm without any changes to the original approach. When trying to follow the original paper's methodology, precision is only 3.59% and recall is 88.00% (see line 2 in Table 3 for details.) In addition to 526 positive examples (out of a total of 600), the classifier returns also 14,194 misclassified examples - incorrect term pairs wrongly classified as correct. We have performed an error analysis and found that almost all incorrectly classified term pairs are cases of partial translation where one unit in a multi-word term has a correct Giza++ dictionary translation in the corresponding term in the other language (Some examples can be seen in Table 2). Such examples accounted for around 82% (11,663) of misclassified term pairs.

4. Adaptation: Experiments for improving the reimplementation

The results in our experiments differ dramatically from the results obtained by Aker et al. (2013). Their approach yields excellent results with perfect precision (100%) and 66% recall for Slovenian. Given that there are 600 positive term pairs in the test set, their results mean that the classifier returns only around 400 term pairs. In contrast, our reimplementation attempts saw the classifier return a lot more of total assigned positive term pairs - 14,720, with 14,194 of them misclassified (false positives).

Feature	Category	Description	Type
isFirstWordTranslated	Dictionary	Checks whether the first word of the source term is a translation of the first word in the target term	Binary
isLastWordTranslated	Dictionary	Checks whether the last word of the source term is a translation of the last word in the target term	Binary
percentageOfTranslatedWords	Dictionary	Ratio of source words that have a translation in the target term	Numeric
percentageOfNotTranslatedWords	Dictionary	Ratio of source words that do not have a translation in the target term	Numeric
longestTranslatedUnitInPercentage	Dictionary	Ratio of the longest contiguous sequence of source words which has a translation in the target term (compared to the source term length)	Numeric
longestNotTranslatedUnitInPercentage	Dictionary	Ratio of the longest contiguous sequence of source words which do not have a translation in the target term (compared to the source term length)	Numeric
Longest Common Subsequence Ratio	Cognate	Measures the longest common non-consecutive sequence of characters between two strings	Numeric
Longest Common Substring Ratio	Cognate	Measures the longest common consecutive string (LCST) of characters that two strings have in common	Numeric
Dice similarity	Cognate	2*LCST / (len(source) + len(target))	Numeric
Needlemann-Wunsch distance	Cognate	LCST / min(len(source), len(target))	Numeric
Normalized Levensthein distance (LD)	Cognate	1 - LD / max(len(source), len(target))	Numeric
isFirstWordCovered	Combination	A binary feature indicating whether the first word in the source term has a translation or transliteration in the target term	Binary
isLastWordCovered	Combination	A binary feature indicating whether the last word in the source term has a translation or transliteration in the target term	Binary
percentageOfCoverage	Combination	Returns the percentage of source term words which have a translation or transliteration in the target term	Numeric
percentageOfNonCoverage	Combination	Returns the percentage of source term words which have neither a translation nor transliteration in the target term	Numeric
difBetweenCoverageAndNonCoverage	Combination	Returns the difference between the last two features	Numeric

Table 1: Features used in the experiments. Note that some features are used more than once because they are direction-dependent.

EN	SL	Giza++
agrarian reform	kmetijski odpadki	agrarian, kmetijski, 0.29737
Brussels region	območje proste trgovine	region, območje, 0.0970153
energy transport	nacionalni prevoz	transport, prevoz, 0.442456
fishery product	tekstilni izdelek	product, izdelek, 0.306948

Table 2: Examples of negative term pairs misclassified as positive. Column 1 contains the English term, column 2 contains the Slovenian term and column 3 contains the Giza++ dictionary entry responsible for positive dictionary-based features.

These results are clearly not useful for our goals, which is why in this section we present several methods aiming at improving the results. To do so, we have taken the following steps:

- Giza++ cleaning
- Lemmatization
- Using only those terms that can be found in the Giza++ training corpora (i.e. DGT)
- Same ratio of positive/negative examples in the training and test set
- Training set selection

4.1. Giza++ cleaning

The output of the Giza++ tool contained a lot of noise and we thought it could perhaps have a detrimental effect on the results. There is no mention of any sophisticated Giza++ dictionary cleaning in the original paper beyond removing all entries where probability is lower then 0.05 and entries where the source word is less than 4 characters and the target word more than 5 characters in length and vice versa. For clean Giza++ dictionaries, we used the resources described in Aker et al. (2014) and available via the META-SHARE repository[4] (Piperidis et al., 2014), specifically, the

[4] http://metashare.tilde.com/repository/browse/probabilistic-

No.	Config	Training set size	Training set pos/neg ratio	Precision	Recall	F-score
1	Reported by (Aker et al., 2013)	12,400	1:1	1	0.6600	0.7900
2	Reimplementation approach	12,966	1:1	0.0359	0.8800	0.0689
3	Giza++ cleaning	12,966	1:1	0.0384	0.7789	0.0731
4	Giza++ cleaning and lemmatization	12,966	1:1	0.0373	0.8150	0.0713
5	Only terms that are in Giza++	8,306	1:1	0.0645	**0.9150**	0.1205
6	Training set 1:200	1,303,083	1:200	0.4299	0.7617	0.5496
7	Training set filtering 1	6,426	1:1	0.5969	0.64167	0.6185
8	Training set filtering 2	35,343	1:10	0.9042	0.5350	**0.6723**
9	Training set filtering 3	645,813	1:200	**0.9342**	0.4966	0.6485

Table 3: Results. No. 1 presents the results reported by the authors, No. 2 our reimplementation of the approach and No.3-9 our modifications of the first reimplementation with the aim of improving the results.

transliteration-based approach which yielded the best results according to the cited paper.
With clean Giza++ dictionaries, precision and F-score improves marginally by less than 0.5% at a cost of a much lower recall (10% lower). For details, see Table 3, line 3.

4.2. Lemmatization

The original paper does not mention lemmatization which is why we assumed that all input data (Giza++ dictionaries, EUROVOC thesaurus) are not lemmatized. They state that to capture words with morphological differences, they don't perform direct string matching but utilize Levenshtein Distance and two words are considered equal if the Levenshtein Distance (Levenshtein, 1966) is equal or higher than 0.95. This led us to believe that no lemmatization was used. Nevertheless, we thought lemmatizing the input data could potentially improve the results which is why we adapted the algorithm to perform lemmatization (using Lemmagen (Juršic et al., 2010)) of the Giza++ dictionary and the EUROVOC terms. We have also removed the Levensthein distance string matching and replaced it with direct string matching (i.e. word A is equal to word B, if word A is exactly the same as B), which drastically improved the execution time of the software.
We considered lemmatization as a factor that could explain the difference in results obtained by us and Aker et al. (2013), but our experiments on lemmatized and unlemmatized clean Giza++ dictionaries show that lemmatization does not have a significant impact on the results. Compared to the configuration with unlemmatized clean Giza++ dictionaries, in the configuration with lemmatized Giza++ dictionaries precision was slightly lower (by 0.1%), recall was a bit higher (by around 4%) and F-score was lower by 0.2%. For details, see Table 3, line 4.

4.3. Using only those terms that can be found in the Giza++ training corpora

We thought that one of the reasons for low results can be that not all EUROVOC terms actually appear in the Giza++ training data (i.e. DGT translation memory). The term translations that do not appear in the Giza++ training data could have dictionary-based features similar to the generated negative examples, which could affect the precision of a classifier that was trained on those terms. We found that only 4,153 out of 7,083 terms of the entire EUROVOC thesaurus do in fact appear in a DGT translation memory. Using only these terms in the classifier training set, did improve the precision to 6.5% and recall to 91.5%. For details, see Table 3, line 5.

4.4. Unbalanced training set

In the original paper, the training set is balanced (i.e. the ratio of positive vs. negative examples is 1) but the test set is not (the ratio is around 1:2000). Since our classifier had low precision and relatively high recall, we figured that an unbalanced training set with much more negative than positive examples could improve the former. To test this, we trained the classifier on an unbalanced training set with a 1:200 ratio of positive vs. negative examples [5] This improved precision of the classifier to 42.99% but reduced recall to 76.16%. Nevertheless, we managed to improve the F-score from 6.9% in the reimplementation approach to 54.9%. For details, see Table 3, line 6.

4.5. Training set filtering

The original paper mentions that their classifier initially achieved low precision on Lithuanian language training set, which they were able to improve by manually removing 467 positive term pairs that had the same characteristics as negative examples from the training set. No manual removal is mentioned for Slovenian.
According to our error analysis, the main problem present partial translations in positive term pairs, where one of the words in the source term has a corresponding translation in the target term. These terms have similar characteristics as a number of generated negative examples, which are consequently classified as false positives. To solve this problem, we focused on the features that would eliminate this partial translations from the training set. After a systematic experimentation, we noticed that we can drastically improve precision if we only keep positive term pairs with the following feature values in the training set:

bilingual-dictionaries-from-dgt-parallel-corpus-for-slovenian-english/fa1e1cb47ef111e5aa3b001dd8b71c66f763b373c00545dfb239b12751e5b339/

[5] 1:200 imbalance ratio was the largest imbalance we tried, since the testing results indicated that no further gains could be achieved by increasing the imbalance even more

- isfirstwordTranslated = True
- islasttwordTranslated = True
- percentageOfCoverage > 0.66
- isfirstwordTranslated-reversed = True
- islasttwordTranslated-reversed = True
- percentageOfCoverage-reversed > 0.66

We managed to improve precision to 59.7% with this approach (see Table 3, line 7.) and when combining it with the previous approach of having an unbalanced training set, we manage to achieve a 90.42% precision and a 53.50% recall, improving the F-score to 67.23% (see Table 3, line 8), when the imbalance ratio was 1:10. With an even more unbalanced training set (1:200), we managed to achieve the best precision of 93.42% at the expanse of a lower recall (49.43%).

5. Manual evaluation

Similar to the original paper, we also performed manual evaluation on a random subset of term pairs classified as positive by the classifier (using the configuration No. 9 that yielded the best results). While the authors of the original approach extract monolingual terms using the term extraction and tagging tool TWSC (Pinnis et al., 2012), we use a terminology extraction workflow described in Vintar (2010) and further expanded in Pollak et al. (2012). Both use a similar approach - terms are first extracted using morphosyntactic patterns and then filtered using statistical measures: TWSC uses pointwise mutual information and TF*IDF, while Vintar (2010) compares the relative frequencies of words composing a term in the domain-specific (i.e. the one we are extracting terminology from) corpus and a general language corpus.

In contrast to the original paper where they extracted terms from domain-specific Wikipedia articles (for the English-German language pair), we are using two translation memories - one containing finance-related content, the other containing IT content. Another difference is that extraction in the original papers was done on comparable corpora, but we extracted terms from parallel corpora - which is why we expected our results to be better. Each source term is paired with each target term (just as in the original paper - if both term lists contained 100 terms, we would have 10,000 term pairs) and extract the features for each term pair. The term pairs were then presented to the classifier that labeled them as correct or incorrect term translations. Afterwards, we took a random subset of 200 term pairs labeled as correct and showed them to an experienced translator[6] fluent in both languages who evaluated them according to the criteria set out in the original paper:

- **1 - Equivalence**: The terms are exact translations/transliterations of each other.
- **2 - Inclusion**: Not an exact translation/transliteration, but an exact translation/transliteration of one term is entirely contained within the term in the other language.
- **3 - Overlap**: Not category 1 or 2, but the terms share at least one translated/transliterated word.
- **4 - Unrelated**: No word in either term is a translation/transliteration of a word in the other.

Domain	1	2	3	4
Reported in Aker et al. (2013)				
IT, Ann. 1	0.81	0.06	0.06	0.07
IT, Ann. 2	0.83	0.07	0.07	0.03
Auto, Ann. 1	0.66	0.12	0.16	0.06
Auto, Ann. 2	0.60	0.15	0.16	0.09
Reimplementation				
Finance	0.72	0.09	0.12	0.07
IT	0.79	0.01	0.09	0.12

Table 4: Manual evaluation results. Ann. stands for "Annotator" since the original paper uses two annotators.

The results of the manual evaluation can be found in Table 4. Manual evaluation showed that 72% of positive term pairs in the Finance domain, and 79% of positive term pairs in the IT domain were correctly classified by the classifier. Compared to the original paper, we believe these results are comparable when taking into account the different monolingual extraction procedures ((Pinnis et al., 2012) vs. (Vintar, 2010)), the different language pairs (English-German vs. English-Slovenian) and the human factor related to different annotators. Note however, that given the fact that we used parallel corpora, we would expect our results to be better.

6. Conclusions and future work

In this paper, we tried to reimplement the approach to bilingual term alignment using machine learning by Aker et al. (2013). They approach term alignment as a bilingual classification task - for each term pair, they create various features based on word dictionaries (i.e. created with Giza++ from the DGT translation memory) and word similarities across languages. They evaluated their classifier on a held-out set of term pairs and additionally by manual evaluation. Their results on the held-out set were excellent, with 100% precision and 66% recall for the English-Slovenian language pair.

Our reimplementation attempt focused just on the English-Slovenian language pair (in contrast with the original article where they had altogether 20 language pairs) and we were unable to replicate the results following the procedures described in the paper. In fact, our results have been dramatically different from the original paper with precision being less than 4% and recall close to 90%. We then tested several different strategies for improving the results ranging from Giza++ dictionary cleaning, lemmatization, different ratios of positive and negative examples in the training and test

[6]The original paper used two annotators, hence two lines for each domain in Table 4

sets, to training set filtering. The last strategy proved to be the most effective - we were able to achieve a precision of almost 91% and a recall of 52% which is closer to the original results reported by the authors of the approach. It is possible that in the original experiments authors performed a similar training set filtering strategy, because the original paper mentions that their classifier initially achieved low precision on Lithuanian language training set, which they were able to improve by manually removing positive term pairs that had the same characteristics as negative examples from the training set. However, no manual removal is mentioned for Slovenian. We have also performed manual evaluation similar to the original paper and reached roughly the same results.

This paper demonstrates some of the obstacles for research reimplementation, such as lack of detail and code unavailability. We believe that in this particular case, the discrepancy in the results could be attributed to the scope of the original paper - with more than 20 languages which is also a demonstration of very impressive approach, it would be impossible to describe procedures for all of them. We weren't able to replicate the results of the original paper, but after developing the optimization approaches describe above over the course of several weeks, we were able to reach a useful outcome at the end. We believe that, when the scope of the paper is broad, providing supplementary material online, and preferably the code, is the only was on assuring complete replicability of results. For this reason, in order to help with any future reimplementation attempts of our paper, we are publishing the code at: `http://source.ijs.si/mmartinc/4real2018`.

In terms of future work, we will continue working on improving the accuracy of the classifier, by incorporating the features derived from the parallel corpora (e.g. co-frequency and other measures, see Baisa et al. (2015)), since our main interest is in aligning terminology from translation memories.

7. Acknowledgements

The authors acknowledge the financial support from the Slovenian Research Agency for research core funding (No. P2-0103). The research was partly supported by the industrial project TermIolar (2015-2017)).

8. Bibliographical References

Aker, A., Paramita, M., and Gaizauskas, R. (2013). Extracting bilingual terminologies from comparable corpora. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 402–411.

Aker, A., Paramita, M. L., Pinnis, M., and Gaizauskas, R. (2014). Bilingual dictionaries for all eu languages. In *LREC 2014 Proceedings*, pages 2839–2845. European Language Resources Association.

Baisa, V., Ulipová, B., and Cukr, M. (2015). Bilingual terminology extraction in sketch engine. In *Ninth Workshop on Recent Advances in Slavonic Natural Language Processing*, pages 61–67.

Baldwin, T. and Tanaka, T. (2004). Translation by machine of complex nominals: Getting it right. In *Proceedings of the Workshop on Multiword Expressions: Integrating Processing*, pages 24–31. Association for Computational Linguistics.

Cao, Y. and Li, H. (2002). Base noun phrase translation using web data and the em algorithm. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7. Association for Computational Linguistics.

Daille, B. and Morin, E. (2005). French-english terminology extraction from comparable corpora. In *International Conference on Natural Language Processing*, pages 707–718. Springer.

Foo, J. (2012). *Computational terminology: Exploring bilingual and monolingual term extraction*. Ph.D. thesis, Linköping University Electronic Press.

Gaizauskas, R., Aker, A., and Yang Feng, R. (2012). Automatic bilingual phrase extraction from comparable corpora. In *24th International Conference on Computational Linguistics*, page 23.

Joachims, T. (2002). *Learning to classify text using support vector machines: Methods, theory and algorithms*. Kluwer Academic Publishers.

Juršic, M., Mozetic, I., Erjavec, T., and Lavrac, N. (2010). Lemmagen: Multilingual lemmatisation with induced ripple-down rules. *Journal of Universal Computer Science*, 16(9):1190–1214.

Levenshtein, V. I. (1966). Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10:707, February.

Macken, L., Lefever, E., and Hoste, V. (2013). Texsis: Bilingual terminology extraction from parallel corpora using chunk-based alignment. *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication*, 19(1):1–30.

Mann, G. S. and Yarowsky, D. (2001). Multipath translation lexicon induction via bridge languages. In *Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies*, pages 1–8. Association for Computational Linguistics.

Nassirudin, M. and Purwarianti, A. (2015). Indonesian-japanese term extraction from bilingual corpora using machine learning. In *Advanced Computer Science*

and Information Systems (ICACSIS), 2015 International Conference on, pages 111–116. IEEE.

Och, F. J. and Ney, H. (2000). A comparison of alignment models for statistical machine translation. In *Proceedings of the 18th conference on Computational linguistics-Volume 2*, pages 1086–1090. Association for Computational Linguistics.

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

Pinnis, M., Ljubešic, N., Stefanescu, D., Skadina, I., Tadic, M., and Gornostay, T. (2012). Term extraction, tagging, and mapping tools for under-resourced languages. In *Proceedings of the 10th Conference on Terminology and Knowledge Engineering (TKE 2012), June*, pages 20–21.

Piperidis, S., Papageorgiou, H., Spurk, C., Rehm, G., Choukri, K., Hamon, O., Calzolari, N., Del Gratta, R., Magnini, B., and Girardi, C. (2014). Meta-share: One year after. In *LREC*, pages 1532–1538.

Pollak, S., Vavpetic, A., Kranjc, J., Lavrac, N., and Vintar, S. (2012). Nlp workflow for on-line definition extraction from english and slovene text corpora. In *KONVENS*, pages 53–60.

Steinberger, R., Pouliquen, B., and Hagman, J. (2002). Cross-lingual document similarity calculation using the multilingual thesaurus eurovoc. *Computational Linguistics and Intelligent Text Processing*, pages 101–121.

Vintar, S. (2010). Bilingual term recognition revisited: The bag-of-equivalents term alignment approach and its evaluation. *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication*, 16(2):141–158.