Good Examples for Terminology Databases in Translation Industry

Andraž Repar[1][2], **Senja Pollak**[3]

[1]Jozef Stefan International Postgraduate School, Jamova 39, 1000 Ljubljana, Slovenia
[2]Iolar d.o.o, Parmova 51, 1000 Ljubljana, Slovenia
[3]Jozef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia
E-mail: repar.andraz@gmail.com, senja.pollak@ijs.si

Abstract

This paper deals with finding good examples for terminology database entries in the translation industry. When extracting terms from bilingual translation memory exchange files, it is very easy to also extract example sentences to showcase the use of the term in practice. However, there are usually a lot of sentences containing the term and selecting an appropriate example is not a straightforward task. In this paper, we explore the use of data mining techniques to find good term examples. After constructing the corpus from a large English-Slovenian bilingual file from a financial domain, we extract linguistic features and load them into the Weka data mining environment to analyze the performance of various classifiers, resulting in 0.8 precision for positive class (good examples) and 0.85 overall accuracy. While the model was tested only on one language combination, the nature of most features is language-independent which suggests that the model could be used successfully for other language combinations.

Keywords: terminology; good example; data mining; classification

1. Introduction

When building bilingual terminology databases with automatic term extraction from translation memory exchange files (TMX[1]), it often makes sense to include an example sentence[2] to see how the term in question behaves in context. But adding just any random sentence is hardly a good strategy – it is imperative that the sentence be as illustrative as possible. However, that is easier said than done. What at first appears to be a relatively straightforward task turns out to be anything but and a more systematic approach has to be taken. According to Kilgarriff et al. (2008), a good dictionary example must be:

- typical, exhibiting frequent and well-dispersed patterns of usage
- informative, helping to elucidate the definition
- intelligible to learners, avoiding gratuitously difficult lexis and structures, puzzling or distracting names, anaphoric references or other deictics which cannot be understood without access to the wider context.

Regarding the extraction of term examples, we can identify two lines of research. While on the one hand, definitions can be considered as optimal examples – with automated methods developed for several languages including English (Navigli & Velardi, 2010), Dutch (Westerhout, 2010), French (Malaisé et al., 2004), German (Storrer & Wellinghoff, 2006), Portuguese (Del Gaudio et al., 2014), and Slovene (Pollak et al., 2012) – other authors focus on extraction of good examples. Our work focuses on good examples, since in translation memories, the definitions are very rare, whereas including (good) examples is a feasible task.

[1] http://www.ttt.org/oscarStandards/tmx/tmx14-20020710.htm

[2] While it is certainly possible to include more than one example, additional examples have only marginal value. At the end of the day, these examples serve only as a supplement to the main part of terminology databases, such as terms and their definitions.

There are several existing related approaches available for good term example extraction. GDEX, a system described in Kilgarriff et al. (2008), lets the user define criteria for good dictionary examples and was designed to help lexicographers with identifying dictionary examples by ranking sentences according to how likely they are to be good candidates. It served as the basis for GDEX for Slovene (Kosem et al., 2011) whose approach was based on experience and arose from the assumption that experienced lexicographers can provide a useful set of heuristics based on their intuition and skills. In order to do so, they have come up with various criteria and then used them to filter out the unsuitable examples. Finally, Ljubešić & Peronja (2015) use a supervised learning approach to finding good dictionary examples in Croatian. They manually rank a set of monolingual example sentences into four categories and then use a regression algorithm. They obtain a precision of around 80 percent on the 10 top-ranked examples.

We take a similar approach but treat this issue as a binary classification problem. First, two domain experts annotated their own part of a large set of sentence pairs as either good or bad examples of the source/target sentence pair where only segments consisting of a sentence annotated as a good example for both languages are considered as positive examples (examples can be seen in Table 1). A set of linguistic features was then extracted to be used in the data mining phase. The features were extracted with Python scripts and the data were then loaded into Weka programming toolkit (Hall et al., 2009) to build a suitable classification model. The goal is to test the performance of various classifiers to try to find the most suitable one for good example selection.

Besides definitions and term examples, (semi-)automatic extraction of other types of knowledge-rich contexts (Meyer, 2001) is of great importance, especially for terminographic purposes. While one would normally look for only one, or at most a few, good term examples, researchers of knowledge-rich concepts are focusing on a larger subset of a corpus containing information that would be valuable to a human for the construction of a knowledge base (Barrière, 2004). Finding good term examples could thus be considered a sub-field of knowledge-rich context discovery.

This paper is structured as follows: Section 2 describes the data and the linguistic features, Section 3 describes the experimental setup, Section 4 describes the results and Section 5 contains the discussion of results, conclusion and plans for future work.

2. Data preparation

The examples in the dataset are from the domain of banking and finance. The data comes from a TMX file which is used by most translation applications to store completed translations – this means that the text is sentence aligned. It contains the source (English) and target (Slovenian) segments along with some metadata (date, translator name, project etc.). As a preliminary step, a monolingual terminology extraction process (adapted from Pollak et al. (2012)) was run on both sides of the TMX file and a subset of the extracted source and target terms was manually aligned.

Both sets of sentences – the source and target sets – were cleaned of various TMX tags, tokenized and POS-tagged (NLTK's Penn Treebank tokenizer and POS-tagger were used for English (Loper & Bird, 2002), whereas for Slovenian, the Penn Treebank tokenizer was again used for tokenization and the open-source Reldi tagger and lemmatizer was used for Slovenian (Ljubešić et al., 2016)). In addition, the Slovenian sentences were

English	Slovenian	
Allocation to (more) defensive stocks was the main detractor as high beta names rallied strongly amid the positive sentiments – though an overweight exposure to **equities** (versus bonds) has partially mitigated on the underperformance.	Razdelitev sredstev (bolj) obrambnim delnicam je najbolj zmanjšala donosnost, ker se je močno izboljšalo razpoloženje vlagateljev do imen z visokim koeficientom beta – čeprav je večja izpostavljenost **lastniškim vrednostnim papirjem** (v nasprotju z obveznicami) delno ublažila slabšo donosnost.	Bad
The resulting portfolio consisted essentially of financial stocks and **equities** from the energy, consumer goods and healthcare sectors.	Portfelj je vključeval predvsem finančne delnice ter **lastniške vrednostne papirje** energetskega, potrošniškega in zdravstvenega sektorja.	Good
d) In addition, deposits may be held and **money-market instruments** may be acquired; their value together with the value of the money-market funds held as defined in letter c), subject to the provisions of letter e), may total a maximum of 15 percent of Sub-Fund assets.	d) Poleg tega je dovoljeno imeti depozite in pridobiti **instrumente denarnega trga**. Njihova skupna vrednost skupaj z vrednostjo skladov denarnega trga v lasti, kot je določeno v točki c), lahko znaša največ 15 odstotkov sredstev podsklada v skladu z določili iz točke e).	Bad
If a Sub-Fund lends securities and **money-market instruments**, the borrower will normally either resell them quickly or has already done so.	Če podsklad posodi vrednostne papirje in **instrumente denarnega trga**, jih posojilojemalec hitro ponovno proda ali pa je to že naredil.	Good
As remuneration for administrative services rendered to the Company in its capacity as Management Company, BNP PAM Lux will receive a maximum annual fee of 0.15 percent calculated on the average of the **net asset values** of the assets of the various sub-funds of the Company for the period for which the fee is payable.	BNP PAM Lux prejme za administrativne storitve, ki jih v funkciji družbe za upravljanje opravlja za družbo, letno nadomestilo največ 0,15 odstotka, izračunano glede na povprečno **čisto vrednost sredstev** različnih podskladov družbe za obdobje, za obdobje, za katerega se plača nadomestilo.	Bad
Any subscription requests received before this closing time will be executed on the basis of the **net asset value** on the Valuation Day.	Zahtevki za vpis, prejeti v tem roku, bodo izvršeni na podlagi **čiste vrednosti sredstev** na obračunski dan.	Good

Table 1: Good/bad examples. The term in question is written in bold style

also lemmatized with the Reldi tagger and lemmatizer in order to facilitate searching for term positions in sentences. The sentences were transformed to the feature vector representation, where the target variable was a nominal variable with YES/NO classes corresponding to good term examples (positive class YES) and bad term examples (NO). For the manual annotation phase, 1,332 example bilingual sentence pairs for various terms were annotated (two professional translators each annotated one half of the examples). Because one sentence can contain multiple terms, individual sentences (sentence pairs) can be used multiple times for different terms. The dataset produced was somewhat imbalanced (962=NO, 370=YES).

The linguistic features extracted as attributes are listed in Table 2.

Altogether, there were five nominal (three of them had binary values, two had multiple nominal values) and 15 numeric attributes. While most of the features were designed to be language-independent, a few target language features were created with a specific characteristic of the target language in mind (e.g. target personal pronouns and target demonstrative pronouns are aimed at the propensity of the Slovenian language for using pronouns instead of repeating full words).

As mentioned above, the target variable was a nominal variable with YES/NO classes.

Short name	Description	Value
SLength	Source sentence length in characters	Numeric
TLength	Target sentence length in characters	Numeric
TLen by SLen	Target length divided by source length	Numeric
STermPos	Position of term in the source sentence	Numeric
TTermPos	Position of term in the target sentence	Numeric
SNoDig	Number of digits in the source sentence	Numeric
TNoDig	Number of digits in the target sentence	Numeric
SNoWeirdChar	Number of weird characters (brackets, asterisks, hyphens, dashes etc.) in the source sentence	Numeric
TNoWeirdChar	Number of weird characters (brackets, asterisks, hyphens, dashes etc.) in the target sentence	Numeric
TPPron	Number of personal pronouns in the target sentence	Numeric
TDPron	Number of demonstrative pronouns in the target sentence	Numeric
NoComma	Number of commas in the target sentence	Numeric
NoFullstop	Number of fullstops in the target sentence	Numeric
SNonInitCapWords	Number of capitalized words not in the initial position in the source sentence	Numeric
TNonInitCapWords	Number of capitalized words not in the initial position in the target sentence	Numeric
SCap_Punc	Checks whether the source sentence starts with a capitalized word and ends with a punctuation mark	Binary
TCap_Punc	Checks whether the target sentence starts with a capitalized word and ends with a punctuation mark	Binary
SPassV	Checks whether the source sentence contains a passive voice form	Binary
TargetCase	Checks the grammatical case of the target term	Nominal
InitWrdType	Checks the word type of the first word of the target sentence	Nominal

Table 2: Extracted features

3. Experimental setup

This section describes the selection of algorithms, feature transformation and feature selection, and presents the evaluation method.

3.1 Algorithms

We tested and compared the following algorithms implemented in the Weka data mining toolkit (Hall et al., 2009):

- Naïve Bayes is a simple probabilistic classifier.
- The J48 decision tree algorithm in Weka is an implementation of the C4.5 decision tree algorithm. It produces a pruned decision tree which offers good visualization of the data.
- The IBk classifier is Weka's implementation of the k-nearest neighbors approach to classification. Classification is performed on the basis of the majority class of k-nearest neighbors.
- JRip implements a propositional rule learner, Repeated Incremental Pruning to Produce Error Reduction (RIPPER). It generates a set of IF rules which provide an easily interpretable description of the data.
- SMO in Weka implements John Platt's sequential minimal optimization algorithm for training a support vector classifier.
- ZeroR is the majority class classifier used as a baseline.

3.2 Discretization

As the dataset contains a mix of numeric and nominal attributes, discretization could potentially prove beneficial. All classifiers (i.e. the implementation of the classifier in Weka) from Section 3.1 by default support numeric attributes, but Weka also offers a separate discretization functionality which was tested to see if it offers any improvements.

Supervised discretization was used. In order to avoid overfitting when using cross-validation (because supervised discretization takes into account class values), we have used the FilterClassifier meta classifier in Weka which allows you to specify a filter (i.e. supervised discretization) and apply it only on the training data leaving the test data untouched.

3.3 Feature selection

Kosem et al. (2011) discovered that some features are more significant than others. We wanted to test that using Weka's feature selection functionality. Specifically, we selected the AttributeSelectedClassifier in Weka which allows you to select the evaluator for feature selection before running the classifier itself. We chose the WrapperSubsetEval evaluator and selected the respective classifier to select the best possible features (e.g. for J48, first feature selection was performed with the J48 classifier, then a J48 model was built using the selected features).

3.4 Evaluation method

The performance of the classifiers was evaluated in the 10-fold cross-validation setting using the following basic measures: accuracy, precision, recall and F-score. Because we normally have several example sentences per term and we only really need one good example to be included in the termbase, the most important measure for our task is the precision of the positive class (i.e. true positive examples vs all classified positive examples).

$$precision = \frac{tp}{tp + fp} \tag{1}$$

4. Results

Since the dataset is imbalanced, it makes sense to compare the performance of the classifier with the ZeroR classifier which classifies all examples in the majority class (i.e. bad example). Apart from Naïve Bayes with the default configuration, all classifiers return an accuracy higher than the ZeroR baseline. The highest accuracy was recorded with the J48 classifier in combination with feature selection (85.21%). For detailed results, see Table 3.

Feature discretization: Naïve Bayes, J48, IBk and SMO have all exhibited improved precision after feature discretization, but this improvement came in the majority of cases at the expense of lower recall. However, we are primarily interested in precision meaning that discretization has a positive influence on the performance of these classifiers for this task. On the other hand, the performance of JRip slightly decreased with discretization.

Parameter fine-tuning: We have experimented with different parameter settings. For J48, different minimum numbers of objects were tested. Figure 1 plots precision as the parameter MinNumObj increases. Tests were performed with and without discretization. Without

	Precision (positive class)	**Recall (positive class)**	**F-score (positive class)**	**Accuracy**
ZeroR	0	0	0	0.7222
Naïve Bayes	0.440	**0.916**	0.595	0.653
cNaïve Bayes with discretization	0.554	0.819	0.661	0.766
Naïve Bayes with feat. selection	0.534	0.632	0.579	0.745
J48 (MinNumObj=2)	0.734	0.686	**0.709**	0.844
J48 (MinNumObj=9)	0.745	0.665	0.703	0.844
J48 with discretization (MinNumObj=2)	0.753	0.568	0.647	0.828
J48 with discretization (MinNumObj=22)	0.770	0.543	0.637	0.828
J48 with feat. selection (MinNumObj=2)	**0.801**	0.622	0.700	**0.852**
SMO	0.644	0.573	0.607	0.794
SMO with discretization	0.700	0.630	0.663	0.822
SMO with feat. selection	0.646	0.562	0.601	0.793
IBk (k=1)	0.635	0.673	0.654	0.802
IBk (k=7)	0.686	0.619	0.651	0.815
IBk with discretization (k=9)	0.732	0.635	0.680	0.834
IBk with feat. selection (k=9)	0.732	0.643	0.685	0.836
JRip	0.738	0.570	0.643	0.824
JRip with feat. discretization	0.735	0.616	0.671	0.832
JRip with feat. selection	0.763	0.576	0.656	0.833

Table 3: Classification results with different algorithms and parameter settings

discretization, the largest precision was achieved with MinNumObj set to 9 (0.745). This setting results in a tree with 29 leaves. With discretization, the best precision was achieved when MinNumObj was set to 22 (0.770). At this point the tree had nine leaves. Discretizing the data allows us to achieve better precision with the added bonus of having fewer leaves which makes the tree easier to interpret. As can be seen in Figure 2, discretization has a positive influence also on the precision of the IBk[3] classifier. The largest precision is achieved with k set to 9 (0.732). Without discretization, precision never breaks the 0.7 barrier. For Naïve Bayes, SMO and JRip, we have not been able to improve considerably the performance of these two classifiers by adjusting the respective parameters and have used the default parameters throughout the analysis.

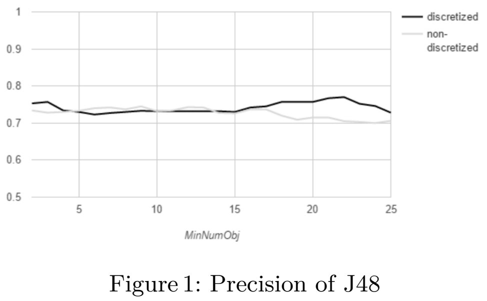


Figure 1: Precision of J48

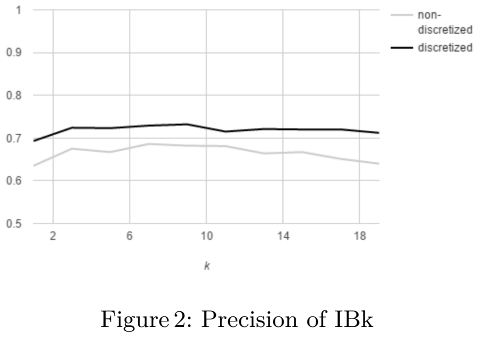


Figure 2: Precision of IBk

Feature selection: We tested the role of feature selection by applying the Weka's feature selection functionality, described in Section 3.3. In terms of precision, feature selection

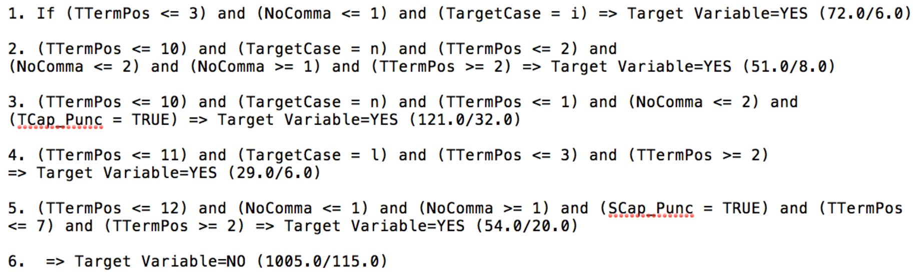
[3] To avoid ties in a binary classification problem, we have only used odd values of k.

improved the precision of all classifiers, but this improvement came at the expense of recall (for details see Table 3). When applying the feature selection, the number of features has fallen considerably for all classifiers (e.g., from all the features seven were selected for J48, and six for JRip), except for SMO where the number of features fell only marginally to 17. In Table 4 the features resulting from the feature selection process are listed. It can be seen that the target term position, number of commas and the source initial capitalization/final punctuation are present in almost all the classification models (in four out of five models).

Classifier	**Selected features**
NaiveBayes	TLength, SCap_Punc, TargetCase
J48	TLength, STermPos, SNoDig, TNoDig, TTermPos, TCap_Punc, NoComma
SMO	SLength, TLength, STermPos, SNoDig, TNoDig, SNoWeirdChar, TNoWeirdChar, SCap_Punc, TtermPos, TCap_Punc, TargetCase, TDPron, NoComma, NoFullstop, InitWrdType, SNonInitCapWrds, TNonInitCapWrds
IBk	TNoWeirdChar, SCap_Punc, TTermPos, TargetCase, NoComma, NoFullstop
JRip	SCap_Punc, TTermPos, TCap_Punc, TargetCase, TPPron, NoComma

Table 4: The most informative features (feature selection results)

Model interpretation: For model interpretability the most interesting results were produced with the JRip classifier which produces a set of easily interpretable rules. In Figure 3 we present the JRip model with the feature selection step. For example, Rule 1 says that for a bilingual sentence pair to be a good term example, the target term has to be positioned within the first four words (the first position is 0) from the beginning of the sentence, there should be only one or no commas in the target sentence and the target term should be in the instrumental case. Interestingly, this rule contains no mention of any source features which could indicate that there is a strong relationship between the source and target sentences (i.e. if a sentence is a good example in one language, its corresponding pair will also be a good example in the other language). This rule has a very high precision (0.917), and covers 72 examples. Subsequently new rules are formed to cover other, still uncovered, instances.



```
1. If (TTermPos <= 3) and (NoComma <= 1) and (TargetCase = i) => Target Variable=YES (72.0/6.0)

2. (TTermPos <= 10) and (TargetCase = n) and (TTermPos <= 2) and
(NoComma <= 2) and (NoComma >= 1) and (TTermPos >= 2) => Target Variable=YES (51.0/8.0)

3. (TTermPos <= 10) and (TargetCase = n) and (TTermPos <= 1) and (NoComma <= 2) and
(TCap_Punc = TRUE) => Target Variable=YES (121.0/32.0)

4. (TTermPos <= 11) and (TargetCase = l) and (TTermPos <= 3) and (TTermPos >= 2)
=> Target Variable=YES (29.0/6.0)

5. (TTermPos <= 12) and (NoComma <= 1) and (NoComma >= 1) and (SCap_Punc = TRUE) and (TTermPos
<= 7) and (TTermPos >= 2) => Target Variable=YES (54.0/20.0)

6.  => Target Variable=NO (1005.0/115.0)
```

Figure 3: JRip rules on the dataset with feature selection.

We also analysed the results without feature selection and compared the results on all features and all features with discretization. We provide the first JRip rule for each. For the representation without discretization and feature selection the first rule (covering 134 instances, out of which 18 are misclassified, leading to the rule precision of 0.866) is the following:

1. (STermPos <= 6) and (TNoWeirdChar <= 0) and (TLength >= 13) and (TTermPos <= 3) and (SLength <= 28) => Target Variable=YES (134.0/18.0)

Again, the term position is important (the source and target term position), and other features are the length of the target and source sentences, and no weird characters in the target sentence.

On the discretized features the first rule (with precision of 0.795) is the following:

1. (TTermPos = '(0.5-3.5]') and (SLength = '(-inf-27.5]') and (SNonInitCapWrds = '(-inf-3.5]') and (SCap_Punc = TRUE) => Target Variable=YES (253.0/52.0)

As in the previous rule, the term position is important, (but here it is just the target term's position that was selected), followed by the requirement for a low number of non-initial capitalized words and the need for the first word in the source sentence to be capitalized and the source sentence to contain a final punctuation mark.

Comparison with GDEX: We can align our findings with the findings of characteristics of good examples for lexicography – the GDEX for Slovene by Kosem et al. (2011).While a direct comparison is not possible due to the different setup (e.g. GDEX only deals with monolingual data and because of the different features involved), there are nevertheless some comparisons to be made. In GDEX, the following features were found to be the most relevant: preferred sentence length, relative keyword position in the sentence, penalty for keyword repetition, penalty for words exceeding the prescribed maximum length, and penalty for sentences exceeding maximum length. As seen in Table 5, some of the most prominent features offered by the feature selection functionality in this paper are similar: source length and target term position are closely related with preferred sentence length and relative keyword position in GDEX. Looking at the values produced by the JRip classifier on the discretized data for these two features, we can observe similarities with GDEX results (see Table 5).

GDEX (Slovene1 configuration)	**Our approach**
Relative keyword position between 0-20% of the sentence	Target term position = '(0.5–3.5]'
Preferred sentence length min 8 and max 30 words	Source length (characters) = '(-inf–27.5]

Table 5: Comparison with GDEX on target term position and sentence length

5. Discussion and conclusions

We presented the data mining experiments on the task of finding good term examples for terminological databases, where the input files are parallel sentences from a translation memory.

Overall, all classifiers apart from Naïve Bayes with the default configuration have provided some level of improvement in accuracy over the ZeroR classifier which classifies all instances into the same class (bad examples) (see Table 3). In terms of precision of the positive class—which is also the most relevant measure for our goal—as well as overall accuracy, the best classifier for this task seems to be J48 (with feature selection and minimum number of objects set to two) and the worst Naïve Bayes – the difference between the highest (J48, 0.801) and the lowest precision (Naïve Bayes, 0.440) is around 50%. Weka's implementations of k-nearest neighbours (IBk), support vector machine (SMO) and JRIP also perform quite well and could be good candidates for future research into good term example extraction. However, it is important to note that SMO is considerably more demanding in terms of processing power and takes much longer to complete, which can be a significant factor for practical applications; it also provides less interpretable results.

Fine-tuning parameters of the classifiers J48 and IBk provided some improvement in performance as well as reducing the leaf count in a J48 decision tree. We were unable to increase the performance of the other three classifiers by fine-tuning their respective parameters.

Supervised discretization has proved to be beneficial for the precision of classifiers with all classifiers (except for JRip) improving their results after discretization. The same holds true for feature selection. In general, the improvements due to feature selection were greater than the improvements due to discretization. Feature selection also considerably reduces the number of significant features with the number ranging from three to seven (out of the 20 available), except for SMO where the number of features remained relatively high even after feature selection.

Finally, the JRip classifier provides a set of easily interpretable rules. Some of these rules have even higher precisions (e.g. 0.917) than individual classifiers.

While the results are promising, there is certainly room for improvement. The obvious route to take would be to explore the combination of discretization and feature selection, because we have seen that both improve the precision of the classifiers. Moreover, having a larger dataset with more diverse data from different domains would most likely improve the ability to apply the model to any domain. We have not tested our classifier on language pairs other than English-Slovenian, but most of the extracted features are language independent which suggests that this classifier could also be used successfully for other language pairs. This is something we plan to test in the future.

Finally, the dataset is complex and treating this issue as a binary classification problem may be too simplistic to accurately reflect the differences between various sentences in the dataset. In the future, it would make sense to repeat the experiment with numeric scores (e.g. five being the best example, one being the worst) instead of YES/NO values which would allow us to test regression algorithms. Moreover, extracting word type sequences would allow us to discover the most typical sentence structures of good examples and including features describing word frequencies in reference and domain-specific corpora would unlock a completely new level of analysis.

This paper is part of a larger research into developing a comprehensive terminology extraction system for a translation service provider. In addition to extracting terms and good term examples, we will focus on other types of information that can be extracted from TMX files, such as definitions, collocations or domains. Having the ability to quickly and accurately extract good term examples would be of great benefit to this system.

6. Acknowledgements

This research was done as part of cooperation between the JSI Institute and Iolar d.o.o. in the scope of the TermIolar project.

7. References

Barrière, C. (2004). *Knowledge-Rich Contexts Discovery.* Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 187–201. URL http://dx.doi.org/10.1007/978-3-540-24840-8_14.

Del Gaudio, R., Batista, G. & Branco, A. (2014). Coping with highly imbalanced datasets: A case study with definition extraction in a multilingual setting. *Natural Language Engineering*, 20(3), p. 327–359.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. & Witten, I.H. (2009). The WEKA Data Mining Software: An Update. *SIGKDD Explor. Newsl.*, 11(1), pp. 10–18.

Kilgarriff, A., Husák, M., McAdam, K., Rundell, M. & Rychlỳ, P. (2008). GDEX: Automatically Finding Good Dictionary Examples in a Corpus. In *Proceedings of the 13th EURALEX International Congress.* Barcelona, Spain: Institut Universitari de Linguistica Aplicada, Universitat Pompeu Fabra, pp. 425–432. URL https://euralex.org/publications/gdex-automatically-finding-good-dictionary-examples-in-a-corpus/.

Kosem, I., Husak, M. & McCarthy, D. (2011). GDEX for Slovene. In *Electronic lexicography in the 21st century: new applications for new users: Proceedings of eLex 2011.* Bled, Slovenia: Trojina, Institute for Applied Slovene Studies, pp. 151–159. URL http://elex2011.trojina.si/Vsebine/proceedings/eLex2011-19.pdf.

Ljubešić, N., Klubička, F., Agić, Ž. & Jazbec, I.P. (2016). New inflectional lexicons and training corpora for improved morphosyntactic annotation of Croatian and Serbian. In *Tenth International Conference on Language Resources and Evaluation.* Portorož, Slovenia: European Language Resources Association, pp. 4264–4270. URL http://www.lrec-conf.org/proceedings/lrec2016/pdf/340_Paper.pdf.

Ljubešić, N. & Peronja, M. (2015). Predicting corpus example quality via supervised machine learning. In *Electronic Lexicography in the 21st Century: Linking Lexical Data in the Digital Age: Proceedings of eLex 2015.* pp. 477–485.

Loper, E. & Bird, S. (2002). NLTK: The Natural Language Toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1.* Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 63–70. URL http://dx.doi.org/10.3115/1118108.1118117.

Malaisé, V., Zweigenbaum, P. & Bachimont, B. (2004). Detecting Semantic Relations between Terms in Definitions. In *COLING 2004 CompuTerm 2004: 3rd International Workshop on Computational Terminology.* Geneva, Switzerland: COLING, pp. 55–62.

Meyer, I. (2001). Extracting knowledge-rich contexts for terminography. *Recent advances in computational terminology*, 2, p. 279.

Navigli, R. & Velardi, P. (2010). Learning word-class lattices for definition and hypernym extraction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics.* Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 1318–1327. URL http://dl.acm.org/citation.cfm?id=1858815.

Pollak, S., Vavpetic, A., Kranjc, J., Lavrac, N. & Vintar, S. (2012). NLP workflow for online definition extraction from English and Slovene text corpora. In *11th Conference on Natural Language Processing, KONVENS 2012, Empirical Methods in Natural Language Processing*. Vienna, Austria: ÖGAI, pp. 53–60. URL http://www.oegai.at/konvens2012/proceedings/10_pollak12o/.

Storrer, A. & Wellinghoff, S. (2006). Automated detection and annotation of term definitions in German text corpora. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC-2006)*. Genoa, Italy: European Language Resources Association (ELRA), pp. 2373–2376. URL http://www.lrec-conf.org/proceedings/lrec2006/pdf/128_pdf.pdf.

Westerhout, E. (2010). *Definition Extraction for Glossary Creation: A Study on Extracting Definitions for Semi-automatic Glossary Creation in Dutch*. International series. LOT.