Ontology-based translation memory maintenance

Andraž Repar
Iolar d.o.o.
Parmova 51 and
Jozef Stefan International Postgraduate School
Jamova 39
1000 Ljubljana, Slovenia
repar.andraz@gmail.com

Senja Pollak
Jozef Stefan Institute
Jamova 39
1000 Ljubljana, Slovenia
senja.pollak@ijs.si

ABSTRACT

In this paper, we explore the use of text mining techniques for translation memory maintenance. Language service providers often have large databases of translations, called translation memories, which have been in use for a long time - leading to a slow population of the translation memory with other domains (i.e. adding financial content to a medical translation memory). To our best knowledge, no tools exist that would effectively separate the content of a translation memory according to different domains. Having the ability to extract individual domains from low-quality translation memories could mean a significant benefit to language service providers looking to utilize modern translation methods, such as machine translation and automated terminology management. In the first stage, we used OntoGen, a semi-automatic ontology building tool which uses text mining techniques, to separate the segments in the translation memory according to domains. In the second stage, we wanted to test whether we could use the domains defined in the previous stage to build classification models - effectively using them as class labels in place of the costly and time-consuming manual annotation of segments.

Keywords

translation memory, language service provider, ontology, OntoGen, text classification

1. INTRODUCTION

In the translation industry, language service providers (LSP) often offer a guarantee to their customers that they will never have to pay twice for the translation of the same text. In order to do so, they have come up with a way of saving and re-using past translations to reduce costs and offer discounts. Starting in the 1970s and 1980s, translation companies began using translation memories which are essentially databases of bilingual segment pairs (source text – target text) along with some metadata. Whenever a new document is received for translation, it is leveraged against the translation memory for "exact" and "fuzzy" matches and the results are used to calculate the final price of the translation. This technology really took off in the 1990s and today virtually every language service provider on the market uses some kind of a translation memory to store translations.

In theory, the translation memory concept involves the use of metadata to clearly mark the segments belonging to different domains and/or customers. However, metadata are often not added due to time pressure or other issues and the information about the domain or customer is lost. Without this information it is difficult to reuse the translation memories for machine translation and/or terminology management. Finally, the quality of a translation memory can also degrade over the years because segments may get accidentally stored in the wrong translation memory (the domain of the segments is not the same as the domain of the translation memory).

In this paper, we analyze one such translation memory used by the translation company Iolar to see whether we could use text mining techniques to extract domains and clean low-quality translation memories. We used OntoGen [4] topic ontology editor to separate the dataset into distinct domains and then used these domains for text classification in Weka [5].

Ontology learning is a well-researched area with researchers using various techniques, such as natural language processing([10]), machine learning ([13]) and information retrieval ([3]). The same can be said of using machine learning for text classification (for example, [11], [9] and [7]). On the other hand, research into using data mining techniques for translation memory maintainance is scarce with most authors focusing on spotting low-quality individual segments. Barbu [1] uses several machine learning algorithms to spot false segment pairs in translation memories, Sabet et al. [6] describes a system for unsupervised cleaning of translation memories without labeled training data based on a configurable and extensible set of filters, and Nahata et al. [8] defines a set of rules for a rule-based classifier which is in turn used to find low-quality segment pairs. A more recent topic that serves a similar purpose is quality estimation of machine translated segments. For example, Specia et al. [12] describe a system that tries to predict the quality of machine translated segments using machine learning.

2. DATA DESCRIPTION

The translation memory analyzed for this article has been in use for almost 15 years and contains parallel translation segments in English and Slovene. Initially, it was meant to store Marketing, Legal and Financial translations, but over the years various other domains have been stored in this translation memory. In addition to the three domains mentioned above, this translation memory also contains a large chunk of IT-related segments, such as user interface strings, user assistance texts and technical documentation of various IT devices (printers, scanners, monitors etc.). Given the content of these documents, we expect to see some overlap between domains – for example, a printer user manual will typically contain some legal information as well as some marketing-like language.

3. EXPERIMENTS

The most obvious way to go about this task would be to manually annotate a dataset from this translation memory and then use it to train a classifier. However, manual annotation is time consuming and costly, so we first utilize OntoGen [4], a semi-automatic and data-driven ontology editor focusing on editing of topic ontologies, and then use the resulting ontology topics for building a text classifier that could be used for other translation memories and documents.

3.1 Preprocessing

The first step involved extracting the segment pairs and filtering them. The Slovene segment parts were discarded because only one language is needed for this task. English was chosen because it is the source language in this translation memory. The TMX file contained 247,103 English-Slovene segment pairs. To cut down on the noise and remove the segments most difficult to classify, we decided to remove all segments with less than 8 words leaving us with 121,593 segments.

3.2 Ontology creation

The selected segments were saved in a Named Line-Documents format suitable for OntoGen. Given the size of the file, the processing in OntoGen was slow-going. We tried various approaches in OntoGen and finally settled on using k-means clustering (with k=10) functionality to generate various sets of segments corresponding to different keywords and then manually group them into meaningful domains based on our translation experience with this translation memory.

After experimenting with various ontology building techniques in OntoGen, the following topic ontology was constructed (followed by the number of documents in parentheses): IT (51,247) (subdivided into ITGeneral and User Interface), Marketing (11,567), Financial (12,987), Legal (42,163) (subdivided into Contracts, Tenders and IT Legal[1]).

A graphical representation is shown in Figure 1.

3.3 Classification

In the final step we exported the domains from OntoGen, attached them to their corresponding segments and loaded the data into Weka machine learning toolkit. We tested various machine learning classification algorithms (Naïve Bayes Multinomial, SVM, J48) to find which one gives the best results. 10-fold cross-validation was used for all experiments. We applied Weka's StringToWordVector filter and used a stoplist (300 most frequent words from the BNC [2] corpus) to filter out the most common words.

[1]This group contains segments from privacy policies and license agreements of various software applications

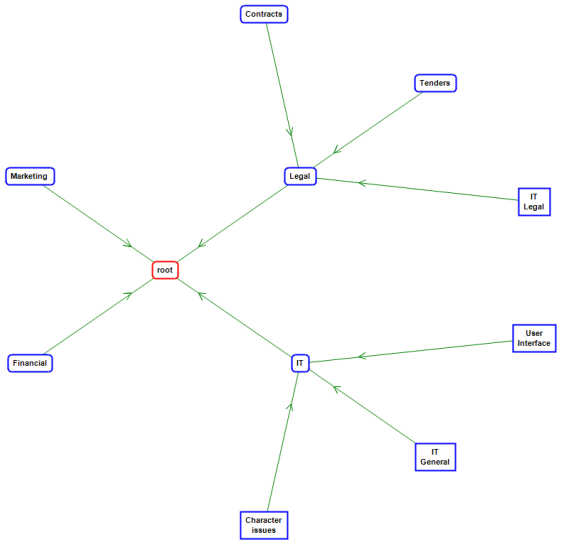

Figure 1: Ontology visualization: 4 main domains are extracted (Financial, Marketing, IT, Legal) with two of them having additional subdomains

In the first phase, we have both topics and subtopics – where a subtopic existed, we glued the topic and subtopic together to get a distinct class. This means we had 7 distinct classes: ITUserInterface, ITITGeneral, Financial, Marketing, LegalContracts, LegalTenders, LegalITLegal.

In the second phase, we used only the main topics – meaning that 4 classes were used: IT, Financial, Marketing, Legal.

Because the original dataset was fairly large (more than 100.000 segments), we had to significantly reduce it in order to be able to complete the calculations in Weka in reasonable time. However, we couldn't just take the first n segments, because the different topics were not uniformly distributed across the dataset. Therefore, we took every 10th segment, leaving us with a dataset of about 10,000 segments.

Tables 1 and 2 contain information about the performance of the three classifiers mentioned in section 3.3. For a detailed analysis see section 4.2.

4. EVALUATION AND INTERPRETATION OF RESULTS

When one evaluates the results of the hierarchical clustering by OntoGen and classification, one should bear in mind that in many cases no clear boundaries between domains exist. This was to be expected on the one hand due to the short length of the documents, and on the other due to the seg-

Table 1: Classifier performance with 7 labels (accuracy of the ZeroR classifier for the majority class = 0.349)

	J48	SMO	NB Multinomial
Accuracy	0.511	0.495	0.583
Precision	0.507	0.483	0.581
Recall	0.511	0.495	0.583
F-measure	0.472	0.483	0.580

Table 2: Classifier performance with 4 labels (accuracy of the ZeroR classifier for the majority class = 0.406)

	J48	SMO	NB Multinomial
Accuracy	0.597	0.619	0.671
Precision	0.615	0.608	0.678
Recall	0.597	0.619	0.671
F-measure	0.576	0.610	0.673

ments that are very difficult to assign to a single domain, for example:

- The system must support operation of the HSM system and the archiving of files even if the file system operates in the Windows cluster.
- The latest Windows operating systems have a firewall built in.

The first sentence comes from a tender document, while the second one comes from an IT user manual. Even for a human annotator, this would be a difficult task and we would most likely see low levels of inter-annotator agreement.

4.1 Ontology creation

To evaluate the results of ontology creation in OntoGen, we extracted 50 random segments for all 7 topics/subtopics, manually annotated them and compared the results.

Overall, a precision of 0.81 is quite good considering that we are working with sentences which are difficult to classify. It is also important to not lose sight of the fact that there can be some overlap between the topics and that certain sentences cannot be adequately classified into any of the available topics. The overlap between the various topics causes a certain degree of ambiguity, but we believe that the precision is high enough to use the topics extracted in OntoGen as class labels for building a classifier.

Table 3: Manual evaluation of the ontology results on 50 segments per domain

Topic	Precision
Financial	0.76
ITGeneral	0.80
ITUserInterface	0.86
LegalContracts	0.80
LegalITLegal	0.86
LegalTenders	0.78
Marketing	0.80
Average	**0.81**

4.2 Classification

The results of the classification with 7 labels are not promising. The performance of all classifiers does exceed the majority class classifier significantly, but the accuracy is not high enough for production use (close to 60% for the best performing classifier). Looking at the confusion matrix in Figure 2, we can observe that the ITGeneral topic overlaps with quite a few other topics and is the largest culprit for the low performance. A significant part of the false positives originate in the ITGeneral topic for all topics apart from Financial and LegalContracts (class c and e in Figure 2). These two topics also have the highest precision (0.688 and 0.668, respectively).

```
=== Confusion Matrix ===

   a    b    c    d    e    f    g   <-- classified as
 236  258   10   24   13   39   25 |    a = ITUserInterface
 197 2512   80  380  137  267  118 |    b = ITITGeneral
   7  131  817   73  231   41   18 |    c = Financial
   7  372   46  511   56  115    9 |    d = Marketing
   9  187  183   52 1321  121  108 |    e = LegalContracts
   9  401   43  150  152  447   38 |    f = LegalTenders
  12  164    8   20   68   34  325 |    g = LegalITLegal
```

Figure 2: Confusion matrix of the Naïve Bayes Multinomial classifier - 7 labels

When we focus only on the main 4 labels, the results are better. Naïve Bayes Multinomial is again the best performing classifier with its accuracy reaching a little over 67%. Looking at the confusion matrix in Figure 3, it is evident that the first 3 labels perform significantly better than the last one. Indeed, the precision of Legal, IT and Financial is around 0.7, while that of Marketing is just a little over 0.4 (for detailed results see Table 4).

```
=== Confusion Matrix ===

   a    b    c    d   <-- classified as
2482  838  261  271 |    a = Legal
 566 3206   80  447 |    b = IT
 268  126  852   72 |    c = Financial
 163  336   50  567 |    d = Marketing
```

Figure 3: Confusion matrix of the Naïve Bayes Multinomial classifier - 4 labels

	Precision	Recall	F-measure
Legal	0.713	0.644	0.677
IT	0.711	0.746	0.728
Financial	0.685	0.646	0.665
Legal	0.418	0.508	0.459

Table 4: Detailed performance of the Naïve Bayes Multinomial classifier

The largest issue that we have not been able to overcome in this analysis is that a huge chunk of the segments in this dataset are IT related – this is especially true of the Marketing and certain Legal segments (e.g. terms of use, privacy

statements or press releases or advertising material for IT devices) which means that it is often difficult to differentiate between a Legal/Marketing segment and a regular IT one. This issue is very clearly seen in the confusion matrix in Figure 3. In contrast, the Financial segments have no immediate relation to any IT content making them a much more distinct category.

5. CONCLUSION AND FUTURE WORK

This paper tries to determine whether text mining techniques can be used to facilitate translation memory maintenance in a language service provider environment. Given the fast-paced nature of work in the translation industry, it is only natural that the quality of translation memories reduce over time. Even if they are perfectly designed, noise will inevitably be introduced leading the reduced usefulness for other language applications.

At the outset, we had two questions: 1) whether OntoGen can be used to divide the content of a particular low-quality translation memory, and 2) whether the resulting topics can be used as labels to build a classifier for other translation memories and documents. The main reason was to find a shortcut for manual annotation which is costly and time-consuming.

We successfully managed to build an ontology, but the boundaries between some topics were relatively vague. One reason for this is that we had to deal with sentences – as opposed to larger chunks of text – which are difficult to classify. The second issue was the fact that many of these topics were in fact inter-related and some of the segments could have easily been classified in more than one domain. In particular, the Legal, IT and Marketing domains are closely related, because a lot of Legal and Marketing segments originated in IT-related translation jobs. One could argue that the IT and Marketing domains could be combined into one category, since there is so much overlap, however from a strictly translator's point of view it makes sense to have separate categories, because different translation strategies are normally used for marketing (i.e. press releases) and general IT (i.e. user manuals, help articles) translation jobs.

The results of the ontology creation were promising with manual evaluation (see Table 3) showing that around 4 in 5 strings were assigned a correct label. However, the picture was much less clear when it came to building a classifier. It turned out that the full ontology was too complex for the classification algorithms used in this paper (see Section 4.2). When we used only the four main topics as labels, the results started approaching acceptable with an accuracy of 67% (compared to 0.406 as majority class). We would still ideally like to see the accuracy breaking the 75% or 80% barrier.

In the current state, the classifier is not accurate enough to be used in production. However, when there are reasonably clear boundaries between topics in OntoGen, the resulting labels can be successfully used – as evident by the performance of the Financial label. This is in itself a useful achievement, because there is currently no way to export just the finance-related segments from the translation memory. An obvious route to better classification performance would be to use just those topics that are clearly separated from the other parts of the dataset.

In terms of future work, we will explore text classification on manually annotated high quality translation memories. Finally, an interesting route would be to utilize domain terminology to enhance highly domain-specific terms assigning higher weight to terminological features.

6. REFERENCES

[1] E. Barbu. Spotting false translation segments in translation memories, 2015.

[2] J. H. Clear. The digital word. chapter The British National Corpus, pages 163–187. MIT Press, Cambridge, MA, USA, 1993.

[3] H. Cunningham. Information extraction, automatic. *Encyclopedia of Language and Linguistics, 2nd Edition*, 5:665–677, 2006.

[4] B. Fortuna, M. Grobelnik, and D. Mladenic. *OntoGen: Semi-automatic Ontology Editor*, pages 309–318. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[5] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18, Nov. 2009.

[6] M. Jalili Sabet, M. Negri, M. Turchi, J. G. C. de Souza, and M. Federico. Tmop: a tool for unsupervised translation memory cleaning. In *Proceedings of ACL-2016 System Demonstrations*, pages 49–54. Association for Computational Linguistics, 2016.

[7] T. Joachims. *Text categorization with Support Vector Machines: Learning with many relevant features*, pages 137–142. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998.

[8] N. Nahata, T. Nayak, S. Pal, and S. Naskar. Rule based classifier for translation memory cleaning. 05 2016.

[9] K. Nigam, A. K. Mccallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using em. *Machine Learning*, 39(2):103–134, May 2000.

[10] S. P. Ponzetto and M. Strube. Deriving a large scale taxonomy from wikipedia. In *Proceedings of the 22Nd National Conference on Artificial Intelligence - Volume 2*, AAAI'07, pages 1440–1445. AAAI Press, 2007.

[11] F. Sebastiani. Machine learning in automated text categorization. *ACM Comput. Surv.*, 34(1):1–47, Mar. 2002.

[12] L. Specia, G. Paetzold, and C. Scarton. Multi-level translation quality prediction with quest++. In *ACL-IJCNLP 2015 System Demonstrations*, pages 115–120, Beijing, China, 2015.

[13] F. M. Suchanek, G. Ifrim, and G. Weikum. Combining linguistic and statistical analysis to extract relations from web documents. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pages 712–717, New York, NY, USA, 2006. ACM.