# An Unsupervised Method for Automatic Translation Memory Cleaning

**Masoud Jalili Sabet**(1), **Matteo Negri**(2), **Marco Turchi**(2), **Eduard Barbu**(3)
(1) School of Electrical and Computer Engineering, University of Tehran, Iran
(2) Fondazione Bruno Kessler, Trento, Italy
(3) Translated srl, Rome, Italy
jalili.masoud@ut.ac.ir
{negri,turchi}@fbk.eu
eduard@translated.net

## Abstract

We address the problem of automatically cleaning a large-scale Translation Memory (TM) in a fully unsupervised fashion, i.e. without human-labelled data. We approach the task by: *i)* designing a set of features that capture the similarity between two text segments in different languages, *ii)* use them to induce reliable training labels for a subset of the translation units (TUs) contained in the TM, and *iii)* use the automatically labelled data to train an ensemble of binary classifiers. We apply our method to clean a test set composed of 1,000 TUs randomly extracted from the English-Italian version of MyMemory, the world's largest public TM. Our results show competitive performance not only against a strong baseline that exploits machine translation, but also against a state-of-the-art method that relies on human-labelled data.

## 1 Introduction

Translation Memories (TMs) are one of the main sources of knowledge supporting human translation with the so-called Computer-assisted Translation (CAT) tools. A TM is a database that stores (*source*, *target*) segments called translation units (TUs). These segments can be sub-sentential fragments, full sentences or even paragraphs in two languages and, ideally, they are perfect translations of each other. Their use in a CAT framework is based on computing a "fuzzy match" score between an input sentence to be translated and the left-hand side (the source) of each TU stored in the TM. If the score is above a certain threshold, the right-hand side (the target) is presented to the user as a translation suggestion. When translating a document with a CAT tool, the user can store each translated (*source*, *target*) pair in the TM for future use. Each newly added TU contributes to the growth of the TM which, as time goes by, will become more and more useful to the user. Due to such constant growth, in which they evolve incorporating users style and terminology, the so-called private TMs represent an invaluable asset for individual translators and translation companies. Collaboratively-created public TMs grow in a less controlled way (e.g. incorporating potentially noisy TUs supplied by anonymous contributors or automatically extracted from the Web) but still remain a practical resource for the translators' community at large.

Together with the *quantity*, the *quality* of the stored material is a crucial factor that determines the usefulness of the TM and, all in all, its value. For this reason, the growth of the TM should go hand in hand with its continuous maintenance. This problem is usually addressed through manual (hence costly) revision, or by applying simple (hence approximate) automatic filtering routines. Advanced automatic methods for tidying up an existing TM would contribute to reduce management costs, increase its quality, speed-up and simplify the daily work of human translators.

Focusing on TM maintenance, we explore an automatic method to clean a large-scale TM by identifying the TUs in which the target is a poor translation of the source. Its main strength is the reliance on a fully unsupervised approach, which makes it independent from the availability of human-labelled data. As it allows us to avoid the burden of acquiring a (possibly large) set of annotated TUs, our method is cost-effective and highly portable across languages and TMs. This contrasts with supervised strategies like the one presented in (Barbu, 2015) or those applied in closely-related tasks such as cross-lingual seman-

*Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 287–292,
Berlin, Germany, August 7-12, 2016. ©2016 Association for Computational Linguistics

| | ENGLISH | ITALIAN<br>*(EN translation)* |
|---|---|---|
| a | traditional costumes of Iceland | costumi tradizionali dell'islanda<br>*(traditional costumes of iceland)* |
| b | Active substances: per dose of 2 ml: | Principi attivi Per ogni dose da 2 ml:<br>*(Active substances Per dose of 2 ml:)* |
| c | The length of time of ... | La durata delperiodo di ...<br>*(The length oftime of ...)* |
| d | ... 4 weeks after administration ... | ... 4 settimane dopo la somministarzione ...<br>*(... 4 weeks after somministarzione ...)* |
| e | 5. ensure the organization of ... | 5.<br>*5.* |
| f | Read package leaflet | Per lo smaltimento leggere il foglio illustrativo<br>*For disposal read the package leaflet* |
| g | beef chuck roast | chuck carne assada<br>*?chuck meat ?assada* |
| h | is an integral part of the contract | risultato della stagione<br>*(result of the season)* |

Table 1: Examples of problematic translation units mined from the English-Italian version of MyMemory.

tic textual similarity,[1] cross-lingual textual entailment (Negri et al., 2013), and quality estimation (QE) for MT (Specia et al., 2009; Mehdad et al., 2012; C. de Souza et al., 2014; Turchi et al., 2014; C. de Souza et al., 2015). Also most of the previous approaches to bilingual data mining/cleaning for statistical MT rely on supervised learning (Resnik and Smith, 2003; Munteanu and Marcu, 2005; Jiang et al., 2009). Unsupervised solutions, like the one proposed by Cui et al. (2013) usually rely on redundancy-based approaches that reward parallel segments containing phrase pairs that are frequent in a training corpus. This idea is well-motivated in the SMT framework but scarcely applicable in the CAT scenario, in which it is crucial to manage and reward rare phrases as a source of useful suggestions for difficult translations.

## 2 The problem

We consider as "problematic TUs" those containing translation errors whose correction during the translation process can reduce translators' productivity. Table 1 provides some examples extracted from the English-Italian training data recently released for the NLP4TM 2016 shared task on cleaning translation memories.[2] As can be seen in the table, TU quality can be affected by a variety of problems. These include: *1.* minor formatting errors like the casing issue in example (a), the casing+punctuation issue in (b) and the missing space in (c), *2.* misspelling errors like the one in (d),[3] *3.* missing or extra words in the translation, as in (e) and (f), *4.* situations in which the translation is awkward (due to mistranslations and/or untranslated terms) like in (g) or it is completely unrelated to the source sentence like in (h).

Especially in the case of collaboratively-created public TMs, these issues are rather frequent. For instance, in the NLP4TM shared task training data (randomly sampled from MyMemory) the instances affected by any of these error types are about 38% of the total.

## 3 Method

Our unsupervised TM cleaning method exploits the independent views of three groups of similarity-based features. These allow us to infer a binary label for a subset of the TUs stored in a large-scale TM. The inferred labels are used to train an ensemble of binary classifiers, specialized to capture different aspects of the general notion of translation quality. Finally, the ensemble of classifiers is used to label the rest of the TM. To minimize overfitting issues, each base classifier exploits features that are different from those used to infer the label of the training instances.

### 3.1 General workflow

Given a TM to be cleaned, our approach consists of two main steps: *i)* label inference and *ii)* training of the base classifiers.

**Label inference.** The first step aims to infer a reliable binary label (1 or 0, respectively for "good" and "bad") for a subset $Z$ of unlabelled TUs randomly selected from the input TM. To this aim, the three groups of features described in §3.2 (say A, B, C) are first organised into combinations of two

[1] http://alt.qcri.org/semeval2016/task1/

[2] http://rgcl.wlv.ac.uk/nlp4tm2016/shared-task/

[3] "*somministARzione*" instead of "*somministRAzione*".

groups (i.e. AB, AC, BC). As the features are different in nature, each combination reflects a particular "view" of the data, which is different from the other combinations.

Then, for each TU in *Z*, we extract the features belonging to each combination. Being designed and normalized to return a similarity score in the [0-1] interval, the result of feature extraction is a vector of numbers whose average value can be computed to sort each TU from the best (avg. close to 1, indicating a high similarity between source and target) to the worst (avg. close to 0). This is done separately for each feature combination, so that the independent views they provide will produce three different ranked lists for the TUs in *Z*.

Finally, the three ranked lists are processed to obtain different sets of positive/negative examples, whose variable size depends on the amount of TUs taken from the top and the bottom of the lists.

**Training of the base classifiers.** Each of the three inferred annotations of *Z* (say $z^1$, $z^2$, $z^3$) reflects the specific view of the two groups of features used to obtain it (i.e. AB for $z^1$, AC for $z^2$, BC for $z^3$). Based on each view, we train a binary classifier using the third group of features (i.e. C for $z^1$, B for $z^2$, A for $z^3$). This results in three base classifiers: $\hat{A}$, $\hat{B}$ and $\hat{C}$ that, in spite of the same shared purpose, are by construction different from each other. This allows us to create an ensemble of base classifiers and to minimize the risk of overfitting, in which we would have incurred by training one single classifier with the same features (A,B,C) used as labelling criterion.

## 3.2 Features

Our features capture different aspects of the similarity between the source and the target of a TU. The degree of similarity is mapped into a numeric score in the [0-1] interval. The full set consists of 31 features, which are organized in three groups.[4]

**Basic features (8).** This group represents a slightly improved variant of those proposed by Barbu (2015). They aim to capture translation quality by looking at surface aspects, such as the possible mismatches in the number of dates, numbers, URLs and XML tags present in the source and target segments.[5] The consistency between the actual source and target languages and those indicated in the TM is also verified. Language identification, carried out with the Langid tool (Lui and Baldwin, 2012), is a highly predictive feature since sometimes the two languages are inverted or even completely different. Other features model the similarity between source and target by computing the direct and inverse ratio between the number of characters and words, as well as the average word length in the two segments. Finally, two features look at the presence of uncommon character or word repetitions.

**QE-derived features (18).** This group contains features borrowed from the closely-related task of MT quality estimation, in which the complexity of the source, the fluency of the target and the adequacy between source and target are modeled as quality indicators. Focusing on the adequacy aspect, we exploit a subset of the features proposed by Camargo de Souza et al. (2013). They use word alignment information to link source and target words and capture the quantity of meaning preserved by the translation. For each segment of a TU, word alignment information is used to calculate: *i)* the proportion of aligned and unaligned word n-grams (n=1,2), *ii)* the ratio between the longest aligned/unaligned word sequence and the length of the segment, *iii)* the average length of the aligned/unaligned word sequences, and *iv)* the position of the first/last unaligned word, normalized by the length of the segment. Word alignment models were trained on the whole TM, using MGIZA++ (Gao and Vogel, 2008).

**Word embeddings (5).** This is a newly developed group of features that rely on cross-lingual word embeddings to identify "good" and "bad" TUs. Cross-lingual word embeddings provide a common vector representation for words in different languages and allow us to build features that look at the same time at the source and target segments. Cross-lingual word embeddings are computed using the method proposed in (Søgaard et al., 2015). Differently from the original paper, which takes advantage of bilingual documents as atomic concepts to bridge the two languages, we use the TUs contained in the whole TM to build the embeddings. Given a TU and a 100-dimensional vector representation of each word in the source and target segments, the new features are: *i)* the cosine similarity between source and

[4] Implemented in TMop: https://github.com/hlt-mt/TMOP

[5] Being these feature very sparse, we collapsed them into a single one, which is set to 1 if any feature has value 1.

target segment vectors obtained by averaging (or using the median) the source and target word vectors; *ii)* the average embedding alignment score obtained by computing the cosine similarity between each source word and all the target words and averaging over the largest cosine score of each source word; *iii)* the average cosine similarity between source/target word alignments; *iv)* a score that merges features *(ii)* and *(iii)* by complementing word alignments (obtained using MGIZA++) with the alignments obtained from word embedding and averaging all the alignment weights.

## 4 Experiments

**Data.** We experiment with the English-Italian version of MyMemory,[6] the world's largest public TM. This collaboratively built TM contains about 11M TUs coming from heterogeneous sources: aggregated private TMs or automatically extracted from the web/corpora, and anonymous contributions of (*source*, *target*) bi-segments. Being large and free, the TM is of great utility for professional translators. Its uncontrolled sources, however, call for accurate cleaning methods (e.g. to make it more accurate, smaller and manageable). From the TM we randomly extracted: *i)* subsets of variable size to automatically obtain training data for the base classifiers and *ii)* a collection of 2,500 TUs manually annotated with binary labels. Data annotation was done by two Italian native speakers properly trained with the same guidelines prepared by the TM owner for periodic manual revisions. After agreement computation (Cohen's kappa is 0.7838), a reconciliation ended up with about 65% positive and 35% negative examples. This pool is randomly split in two parts. One (1,000 instances) is used as test set for our evaluation. The other (1,500 instances) is used to replicate the approach of Barbu (2015) used as term of comparison.

**Learning algorithm.** Our base classifiers are trained with the Extremely Randomized Trees algorithm (Geurts et al., 2006), optimized using 10-fold cross-validation in a randomized search process and combined in a majority voting schema.

**Evaluation metric.** To handle the imbalanced (65%-35%) data distribution, and equally reward the correct classification on both classes, we evaluate performance in terms of balanced accuracy (BA), computed as the average of the accuracies on the two classes (Brodersen et al., 2010).

**Terms of comparison.** We evaluate our approach against two terms of comparison, both stronger than the trivial random baseline achieving a BA of 50.0%. The first competitor (`MT-based`) is a translation-based solution that exploits Bing translator[7] to render the source segment of a TU in the same language of the target. Then, the similarity between the translated source and the target segment is measured in terms of Translation Edit Rate (TER (Snover et al., 2006)). The TU is marked as "good" if the TER is smaller than 0.4 ("bad" otherwise). This value is chosen based on the findings of Turchi et al. (2013), which suggests that only for TER values lower than 0.4 human translators consider MT suggestions as good enough for being post-editable. In our scenario we hence assume that "good" TUs are those featuring a small TER distance between the target and an automatic translation of the source.

The second competitor (`Barbu15`) is the supervised approach proposed by Barbu (2015), which leverages human-labelled data to train an SVM binary classifier. To the best of our knowledge, it represents the state-of-the-art in this task.

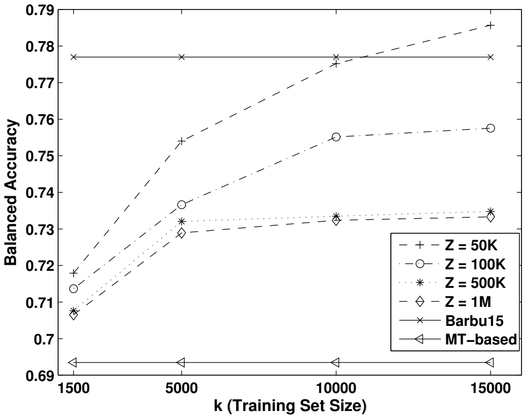

Figure 1: BA results as a function of *Z* and *k*.

## 5 Results and Discussion

The result of the "label inference" step described in §3.1 is a set of automatically labelled TUs to train the base classifiers. Positive and negative examples are respectively the top and the bottom *k* elements extracted from a list of TUs (of size *Z*) ranked according to the inferred similarity between source and target. In this process, the size

[6] https://mymemory.translated.net/

[7] https://www.bing.com/translator/

of the list and the value of $k$ clearly have influence on the separability between the training instances belonging to the two classes. Long lists and small values of $k$ will result in highly polarized training data, with a very high similarity between the instances assigned to each class and feature values respectively close to 1 and 0. Vice-versa, short lists and large values of $k$ will result in less separable training data, with higher variability in the points assigned to each class and in the respective feature values. In light of this trade-off, we analyse performance variations as a function of: *i)* the amount ($Z$) of data considered to initialise the label inference step, and *ii)* the amount ($k$) of training instances used to learn the base classifiers. For the first dimension, we consider four values: 50K (a value compatible with the size of most of the existing TMs), 100K, 500K and 1M units (a value compatible only with a handful of large-scale TMs). For the second dimension we experiment with four balanced training sets, respectively containing: 1.5K (the same amount used in (Barbu, 2015)), 5K, 10K and 15K instances.

Figure 1 illustrates the performance of our TM cleaning method for different values of $Z$ and $k$. Each of the four dashed learning curves refers to one of the four chosen values of $Z$. BA variations for the same line are obtained by increasing the number of training instances $k$ and averaging over three random samples of size $Z$. As can be seen from the figure, the results obtained by our classifiers trained with the inferred data always outperform the `MT-based` system and, in one case ($Z$=50K, $k$=15K), also the `Barbu15` classifier trained with human labelled data.[8] Considering that all our training data are collected without any human intervention, hence eliminating the burden and the high costs of the annotation process, this is an interesting result.

Overall, for the same value of $k$, smaller values of $Z$ consistently show higher performance. At the same time, for the same value of $Z$, increasing $k$ consistently yields higher results. Such improvements, however, are less evident when the pool of TUs used for the label inference step is larger ($Z$>100K). These observations confirm the intuition that classifiers' performance is highly influenced by the relation between the amount and the polarization of the training data. Indeed, looking at the average feature values used to infer the positive and negative instances, we noticed that, for the considered values of $k$, these scores are closer to 0 and 1 for the 1M curve than for the 50K curve. In the former case, highly polarized training data limit the generalisation capability of the base classifiers (and their ability, for instance, to correctly label the borderline test instances), which results in lower BA results.

Nevertheless, it's worth remarking that our larger value of $k$ (15K) represents $30\%$ of the data in the case of $Z$=50K, but just $1.5\%$ of the data in case of $Z$=1M. This suggests that for large values of $Z$, more training points would be probably needed to introduce enough variance in the data and improve over the almost flat curves shown in Figure 1. Exploring this possibility was out of the scope of this initial analysis but would be doable by applying scalable algorithms capable to manage larger quantities of training data (up to 300K, in the case of $Z$=1M). For the time being, a statistically significant improvement of $\sim$1 BA point over a supervised method in the most normal conditions ($Z$=50K) is already a promising step.

## 6 Conclusion

We presented a fully unsupervised method to remove useless TUs from a large-scale TM. Focusing on the identification of wrongly translated segments, we exploited the independent views of different sets of features to: *i)* infer a binary label for a certain amount of TUs, and *ii)* use the automatically labelled units as training data for an ensemble of binary classifiers. Such independent labelling/training routines exploit the "wisdom of the features" to bypass the need of human annotations and obtain competitive performance. Our results are not only better than a strong MT-based baseline, but they also outperform a state-of-the-art approach relying on human-labelled data.

## Acknowledgments

This work has been partially supported by the EC-funded projects ModernMT (H2020 grant agreement no. 645487) and EXPERT (FP7 grant agreement no. 317471). The work carried out at FBK by Masoud Jalili Sabet was sponsored by the EAMT Summer internships 2015 program. The authors would like to thank Anders Søgaard for sharing the initial version of the code for computing word embeddings.

[8] Improvements are statistically significant with $\rho < 0.05$, measured by approximate randomization (Noreen, 1989).

## References

Eduard Barbu. 2015. Spotting False Translation Segments in Translation Memories. In *Proceedings of the Workshop Natural Language Processing for Translation Memories*, pages 9–16, Hissar, Bulgaria, September.

Kay Henning Brodersen, Cheng Soon Ong, Klaas Enno Stephan, and Joachim M. Buhmann. 2010. The Balanced Accuracy and Its Posterior Distribution. In *Proceedings of the 2010 20th International Conference on Pattern Recognition*, ICPR '10, pages 3121–3124, Istanbul, Turkey, August.

José G. C. de Souza, Jesús González-Rubio, Christian Buck, Marco Turchi, and Matteo Negri. 2014. FBK-UPV-UEdin participation in the WMT14 Quality Estimation shared-task. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 322–328, Baltimore, Maryland, USA, June.

José G. C. de Souza, Matteo Negri, Elisa Ricci, and Marco Turchi. 2015. Online Multitask Learning for Machine Translation Quality Estimation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 219–228, Beijing, China, July.

José Guilherme Camargo de Souza, Christian Buck, Marco Turchi, and Matteo Negri. 2013. FBK-UEdin Participation to the WMT13 Quality Estimation Shared Task. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 352–358, Sofia, Bulgaria, August.

Lei Cui, Dongdong Zhang, Shujie Liu, Mu Li, and Ming Zhou. 2013. Bilingual data cleaning for smt using graph-based random walk. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 340–345, Sofia, Bulgaria, August.

Qin Gao and Stephan Vogel. 2008. Parallel implementations of word alignment tool. In *In Proceedings of the ACL 2008 Software Engineering, Testing, and Quality Assurance Workshop*, pages 49–57, Columbus, Ohio, USA, June.

Pierre Geurts, Damien Ernst, and Louis Wehenkel. 2006. Extremely randomized trees. *Machine learning*, 63(1):3–42.

Long Jiang, Shiquan Yang, Ming Zhou, Xiaohua Liu, and Qingsheng Zhu. 2009. Mining bilingual data from the web with adaptively learnt patterns. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 870–878, Suntec, Singapore, August.

Marco Lui and Timothy Baldwin. 2012. langid.py: An Off-the-shelf Language Identification Tool. In *Proceedings of the ACL 2012 System Demonstrations*, pages 25–30, Jeju Island, Korea, July.

Yashar Mehdad, Matteo Negri, and Marcello Federico. 2012. Match without a Referee: Evaluating MT Adequacy without Reference Translations. In *Proceedings of the Machine Translation Workshop (WMT2012)*, pages 171–180, Montréal, Canada, June.

Dragos Stefan Munteanu and Daniel Marcu. 2005. Improving machine translation performance by exploiting non-parallel corpora. *Computational Linguistics*, 31(4):477–504, December.

Matteo Negri, Alessandro Marchetti, Yashar Mehdad, Luisa Bentivogli, and Danilo Giampiccolo. 2013. Semeval-2013 Task 8: Crosslingual Textual Entailment for Content Synchronization. In *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval 2013)*, pages 25–33, Atlanta, Georgia, USA, June.

Erik W. Noreen. 1989. Computer-intensive methods for testing hypotheses: an introduction. *Wiley Interscience*.

Philip Resnik and Noah A. Smith. 2003. The web as a parallel corpus. *Computational Linguistics*, 29(3):349–380.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. In *Proceedings of Association for Machine Translation in the Americas*, pages 223–231, Cambridge, Massachusetts, USA, August.

Anders Søgaard, Željko Agić, Héctor Martínez Alonso, Barbara Plank, Bernd Bohnet, and Anders Johannsen. 2015. Inverted indexing for cross-lingual NLP. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1713–1722, Beijing, China, July.

Lucia Specia, Nicola Cancedda, Marc Dymetman, Marco Turchi, and Nello Cristianini. 2009. Estimating the Sentence-level Quality of Machine Translation Systems. In *Proceedings of the 13th Annual Conference of the European Association for Machine Translation (EAMT-2009)*, pages 28–35, Barcelona, Spain.

Marco Turchi, Matteo Negri, and Marcello Federico. 2013. Coping with the Subjectivity of Human Judgements in MT Quality Estimation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 240–251, Sofia, Bulgaria, August.

Marco Turchi, Antonios Anastasopoulos, José G. C. de Souza, and Matteo Negri. 2014. Adaptive Quality Estimation for Machine Translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 710–720, Baltimore, Maryland, USA, June.