Towards Personalised Simplification based on L2 Learners' Native Language

Alessio Palmero Aprosio[†], **Stefano Menini**[†], **Sara Tonelli**[†]
Luca Ducceschi[‡], **Leonardo Herzog**[‡]
[†]FBK, [‡]University of Trento
{aprosio,menini,satonelli@fbk.eu}
luca.ducceschi@unitn.it
leonardo.herzog@studenti.unitn.it

Abstract

English. We present an approach to improve the selection of complex words for automatic text simplification, addressing the need of L2 learners to take into account their native language during simplification. In particular, we develop a methodology that automatically identifies 'difficult' terms (i.e. false friends) for L2 learners in order to simplify them. We evaluate not only the quality of the detected false friends but also the impact of this methodology on text simplification compared with a standard frequency-based approach.

Italiano. *In questo contributo presentiamo un approccio per selezionare le parole complesse da semplificare in modo automatico, tenendo conto della lingua madre dell'utente. Nello specifico, la nostra metodologia identifica i termini 'difficili' (falsi amici) per l'utente per proporne la semplificazione. In questo contesto, viene valutata non soltanto la qualità dei falsi amici individuati, ma anche l'impatto che questa semplificazione personalizzata ha rispetto ad approcci standard basati sulla frequenza delle parole.*

1 Introduction

The task of automated text simplification has been investigated within the NLP community for several years with a number of different approaches, from rule-based ones (Siddharthan, 2010; Barlacchi and Tonelli, 2013; Scarton et al., 2017) to supervised (Bingel and Søgaard, 2016; Alva-Manchego et al., 2017) and unsupervised ones (Paetzold and Specia, 2016), including recent studies using deep learning (Zhang and Lapata, 2017; Nisioi et al., 2017). Nevertheless, only recently researchers have started to build simplification systems that can *adapt* to users, based on the observation that the preceived simplicity of a document depends a lot on the user profile, including not only specific disabilities but also language proficiency, age, profession, etc. Therefore in the last few months the first approaches to personalised text simplification have been proposed at major conferences, with the goal of simplifying a document for different language proficiency levels (Scarton and Specia, 2018; Bingel et al., 2018; Lee and Yeung, 2018).

Along this research line, we present in this paper an approach to perform automated lexical simplification for L2 learners, able to adapt to the user mother tongue. To our knowledge, this is the first work taking into account this aspect and presenting a solution that, given an Italian document and the user's mother tongue as input, selects only the words that the user may find difficult given his/her knowledge of another language. Specifically, we detect and simplify automatically the terms that may be misleading for the user because they are *false friends*, while we do not simplify those that have an orthographically and semantically similar translation in the user native language (so-called *cognates*). In multilingual settings, for instance while teaching, learning or translating a foreign language, these two phenomena have proven to be very relevant (Ringbom, 1986), because the lexical similarities between the two languages in contact have proven to create interferences, favouring or hindering the course of learning.

We compare our approach to the selection of words to be simplified with a standard frequency-based one, in which only the terms that are not listed in De Mauro's Dictionary of Basic Italian[1] are simplified, regardless of the user native

[1] https://dizionario.internazionale.it/

language. Our experiments are evaluated on the Italian-French pair, but the approach is generic.

2 Approach description

Given a document D_i to be simplified, and a native language L_1 spoken by the user, our approach consists of the following steps:

1. **Candidate selection**: for each content word[2] w_i in D_i, we automatically generate a list of words $W_1 \subset L_1$ which are orthographically similar to w_i. In this phase, several orthographical similarity metrics are evaluated. We keep the 5 most-similar terms to w_i.

2. **False friend and cognate detection**: for each of the 5 most similar words in W_1, we classify whether it is a false friend of w_i or not.

3. **Simplification choice**: Based on the output of the previous steps, the system marks w_i as difficult to understand for the user if there are corresponding false friends in L_1. Otherwise, w_i is left in its original form. When a word is marked as difficult, a subsequent simplification module (not included in this work) should try to find an alternative form (such as a synonym, or a description) to make the term more understandable to the user.

2.1 Candidate Selection

A number of similarity metrics have been presented in the past to identify candidate cognates and false friends, see for example the evaluation in Inkpen and Frunza (2005). We choose three of them, motivated by the fact that we want to have at least one ngram-based metric (XXDICE) and one non ngram-based (Jaro/Winkler). To that, we add a more standard metric, Normalized Edit Distance (NED). The three metrics are explained below:

- XXDICE (Brew et al., 1996). It takes in consideration the shared number of extended bigrams[3] and their position relative to two strings S_1 and S_2. The formula is:

$$\mathrm{XX}(S_1, S_2) = \frac{\sum_B \frac{2}{1+(\mathrm{pos}(x)-\mathrm{pos}(y))^2}}{\mathrm{xb}(S_1) + \mathrm{xb}(S_2)}$$

  where B is the set of pairs of shared extended bigrams (x, y), x in S_1 and y in S_2. The functions $\mathrm{pos}(x)$ and $\mathrm{xb}(S)$ return the position of extended bigram x and the number of extended bigrams in string S respectively.

- NED, Normalized Edit Distance (Wagner and Fischer, 1974). A regular Edit Distance calculates the orthographic difference between two strings assigning a cost to any minimum number of edit operations (deletion, substitution and insertion, all with cost of 1) needed to make them equal. NED is obtained by dividing the edit cost by the length of the longest string.

- Jaro/Winkler (Winkler, 1990). The Jaro similarity metric for two strings S_1 and S_2 is computed as follows:

$$J(S_1, S_2) = \frac{1}{3} \cdot \left(\frac{m}{|S_1|} + \frac{m}{|S_2|} + \frac{m-T}{m} \right)$$

  where m is the number of characters in common, provided that they occur in the same (not interrupted) sequence, and T is the number of transpositions of character in S_1 to obtain S_2. The Winkler variation of the metric adds a bias if the two strings share a prefix.

$$JW(S_1, S_2) = J(S_1, S_2) + (1 - J(S_1, S_2))lp$$

  where l is the number of characters of the common prefix of the two strings, up to four, and p is a scaling factor, usually set to 0.1.

Each of these three measures has some disadvantages. For example, we found that Jaro/Winkler metric boosts the similarity of words with the same root. On the other hand, applying NED leads to several pairs of words having the same similarity score. As a result, two words that are close according to a metric can be far using another metric. To overcome this limitation, we balance the three metrics by computing a weighted average of the three scores tuned on a training set. For details, see Section 3.

nuovovocabolariodibase

[2] Content words are words that have a meaning such as names, adjectives, verbs and adverbs. To extract this information, we use the POS tagger included in the Tint pipeline (Aprosio and Moretti, 2018).

[3] An extended bigram is an ordered letter pair formed by deleting the middle letter from any three letter substring of the word.

2.2 False Friend and Cognate Detection

As for false friend and cognate detection, we rely on a SVM-based classifier and train it on a single feature obtained from a multilingual embedding space (Mikolov et al., 2013), where the user language L_1 and the language of the document to be simplified L_2 are aligned. In particular, the feature is the cosine distance between the embeddings of a given content word w_i in the language L_2 and the embedding of its candidate false friends or cognates in L_1. The intuition behind this approach is that two cognates have a shared semantics and therefore a high cosine similarity, as opposed to false friends, whose meanings are generally unrelated. While past approaches to false friend and cognate detection have already exploited monolingual word embeddings (St Arnaud et al., 2017), we employ for our experiments a multilingual setting, so that the semantic distance between the candidate pairs can be measured in their original language without a preliminary translation.

3 Experimental Setup

In our experiments, we consider a setting in which French speakers would like to make Italian documents easier for them to read. Nevertheless, the approach can be applied to any language pair, given that it requires minimal adaptation.

In order to tune the best similarity metrics combination and to train the SVM classifier, a linguist has manually created an Italian-French gold standard, containing pairs of words marked as either cognates or false friends. These terms were collected from several lists available on the web. Overall, the Ita-Fr dataset contains a training set of 1,531 pairs (940 cognates and 591 false friends) and a test set of 108 pairs (51 cognates and 57 false friends).

For the **candidate selection** step, the goal is to obtain for each term w_i in Italian, the 5 French terms with the highest orthographic similarity. Therefore, given w_i, we compute its similarity with each term in a French online dictionary[4] (New, 2006) using the three scores described in the previous section. The lemmas were normalized for accents and diacritics, in order to avoid poor results of the metrics in cases like *général* and *generale*, where the accented *é* character would be considered different with respect to *e*.[5]

In order to identify the best way to combine the three similarity metrics detailed in Section 2.1., we compute all the possibile combinations of weights on 10 groups of 200 word pairs randomly extracted from the 1,531 pairs in the training set, and then keep the combination that scores the highest average similarity.

In Table 1 we report the percentage of times in which the cognate or false friend of w_i in the training set would appear among the 5 most-similar terms extracted from the French online dictionary according to the three different scores in isolation: XX for XXDICE, JW for Jaro/Winkler and NED for Normalized Edit Distance. We also report the best configuration of the three metrics with the corresponding weight to maximise the presence of a cognate or false friend among the 5 most similar terms. We observe that, while the three metrics in isolation yield a similar result, combining them effectively increases the presence of cognates and false friends among the top candidates. This confirms that the metrics capture three different types of similarity, and that it is recommended to take them all into account when performing candidate selection: an approach where evey metric contributes to detecting false friend / cognate candidates outperforms the single metrics.

XX	JW	NED	% Top 5
1.0	-	-	64.6
-	1.0	-	65.6
-	-	1.0	65.9
0.2	0.4	0.4	77.3

Table 1: Analysis of the candidate selection strategy using different metrics in isolation and in combination.

For **false friends and cognates detection**, we proceed as follows. Given a word w_i in Italian, we identify the 5 most similar words in French using the 0.2-0.4-0.4 score introduced before. In case of ties in the 5th positon, we extend the selection to all the candidates sharing the same similarity value.

Each word pair including w_i and one of the 5 most similar words is then classified as false friend or cognate with a SVM using a radial kernel trained on the 1,531 word pairs in the training set. For the multilingual embeddings used to compute

[4] http://www.lexique.org/

[5] For example, NED between *général* and *generale* returns 0.375 when the two strings are not normalized and 0.125 when they are.

the semantic similarity between the Italian words and their candidates, we use the vectors from Bojanowski et al. (2016)[6] trained on Wikipedia data with fastText (Joulin et al., 2016). We chose these resources since they are available both for Italian and French (and several other languages). For the alignement of the semantic spaces of the two languages we use 22,767 Italian-French word pairs collected from an online dictionary.[7]

4 Evaluation

We perform two types of evaluation. In the first one, the goal is to assess whether the system can correctly identify false friends and cognates in a text. In the second one, we want to check what is the difference between the terms simplified by a system with our approach compared with a standard frequency-based simplification system.

For the first evaluation, we manually create a set of 108 Italian sentences containing one false friend or cognate for French speakers taken from the test set. On each term, we run our algorithm and we consider a term a false friend according to two strategies: *a)* if all 5 most similar words in French are classified as false friends, or *b)* if the majority of them are classified as false friends. Results are reported in Table 2.

	P	R	F1
false friends *(a)*	0.75	0.44	0.55
false friends *(b)*	0.57	0.88	0.69

Table 2: False friends classification using setting *(a)* and *(b)*

The evaluation shows that the two settings lead to two different outcomes. In general terms, the first strategy is more conservative and favours Precision, while the second boosts Recall and F1.

As for the second evaluation, on the same set of sentences, we run our algorithm again, this time trying to classify any content word as being a false friend for French speakers or not. We evaluate this component as being part of a simplification system that simplifies only false friends, and we compare this choice with a more standard approach, in which only 'unusual' or 'unfrequent' terms are simplified. This second choice is taken by comparing each content word with De Mauro's Dictionary of Basic Italian and simplifying only those that are not listed among the 7,000 entries of the basic vocabulary.

This evaluation shows that out of 1,035 content words in the test sentences, our simplification approach based on *a)* would simplify 367 words, and 823 if we adopt the strategy *b)*. Based on De Mauro's dictionary, instead, 240 terms would be simplified. Furthermore, there would be only 76 terms simplified using both strategy *a)* and De Mauro's list, and 154 overlaps for strategy *b)*. This shows that the two approaches are rather complementary and based on different principles. This is evident also looking at the evaluated sentences: while considering frequency lists like De Mauro's, terms such as *accademico* and *speleologo* should be simplified because they are not frequently used in Italian, our approach would not simplify them because they have very similar French translations (*académique* and *spéléologue* respectively), and are not classified as false friends by the system. On the other hand, *vedere* would not be simplified in a standard frequency-based system because it is listed among the 2,000 fundamental words in Italian. However, our approach would identify it as a false friend to be simplified because *vider* in French (transl. *svuotare*) is orthographically very similar to *vedere* but has a completely different meaning.

5 Conclusions

In this work, we have presented an approach supporting personalized simplification in that it enables to adapt the selection of difficult words for lexical simplification to the native language of L2 learners. To our knowledge, this is the first attempt to deal with this kind of adaptation. The approach is relatively easy to apply to new languages provided that they have a similar alphabet, since multilingual embeddings are already available and lists of cognates and false friends, although of limited size, can be easily retrieved online.[8]

The work will be extended along different research directions: first, we will evaluate the approach on other language pairs. Then, we will add a lexical simplification module selecting only the words identified as complex by our approach. For

[6] https://github.com/facebookresearch/fastText/blob/master/pretrained-vectors.md

[7] http://dizionari.corriere.it/

[8] See for example the Wiktionary entries at https://en.wiktionary.org/wiki/Category:False_cognates_and_false_friends

this, we can rely on existing simplification tools (Paetzold and Specia, 2015), which could be tuned to adapt also the simplification choices to the user native language, for example by changing the candidate ranking algorithm. Finally, it would be interesting to involve L2 learners in the evaluation, with the goal to measure the effectiveness of different simplification strategies in a real setting.

Acknowledgments

This work has been supported by the European Commission project SIMPATICO (H2020-EURO-6-2015, grant number 692819). We would like to thank Francesca Fedrizzi for her help in creating the gold standard.

References

Fernando Alva-Manchego, Joachim Bingel, Gustavo Paetzold, Carolina Scarton, and Lucia Specia. 2017. Learning how to simplify from explicit labeling of complex-simplified text pairs. In Greg Kondrak and Taro Watanabe, editors, *Proceedings of the Eighth International Joint Conference on Natural Language Processing, IJCNLP 2017, Taipei, Taiwan, November 27 - December 1, 2017 - Volume 1: Long Papers*, pages 295–305. Asian Federation of Natural Language Processing.

Alessio Palmero Aprosio and Giovanni Moretti. 2018. Tint 2.0: an all-inclusive suite for nlp in italian. In *Proceedings of the Sixth Italian Conference on Computational Linguistics (CLiC-it 2018)*, Torino, Italy.

Gianni Barlacchi and Sara Tonelli. 2013. ERNESTA: A Sentence Simplification Tool for Children's Stories in Italian. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing: 14th International Conference, CICLing 2013, Samos, Greece, March 24-30, 2013, Proceedings, Part II*, pages 476–487, Berlin, Heidelberg. Springer Berlin Heidelberg.

Joachim Bingel and Anders Søgaard. 2016. Text simplification as tree labeling. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 337–343. Association for Computational Linguistics.

Joachim Bingel, Gustavo Paetzold, and Anders Søgaard. 2018. Lexi: A tool for adaptive, personalized text simplification. In *Proceedings of COLING*. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *CoRR*, abs/1607.04606.

Chris Brew, David McKelvie, et al. 1996. Word-pair extraction for lexicography. In *Proceedings of the 2nd International Conference on New Methods in Language Processing*, pages 45–55.

Diana Inkpen and Oana Frunza. 2005. Automatic identification of cognates and false friends in french and english. In *Proceedings of RANLP*, pages 251–257, 01.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

John Lee and Chak Yan Yeung. 2018. Personalizing lexical simplification. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 224–232. Association for Computational Linguistics.

Tomas Mikolov, Quoc V Le, and Ilya Sutskever. 2013. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*.

Boris New. 2006. Lexique 3: Une nouvelle base de données lexicales. In *Actes de la Conférence Traitement Automatique des Langues Naturelles (TALN 2006)*.

Sergiu Nisioi, Sanja Stajner, Simone Paolo Ponzetto, and Liviu P. Dinu. 2017. Exploring neural text simplification models. In Regina Barzilay and Min-Yen Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 2: Short Papers*, pages 85–91. Association for Computational Linguistics.

Gustavo Paetzold and Lucia Specia. 2015. Lexenstein: A framework for lexical simplification. In *ACL-IJCNLP 2015 System Demonstrations*, ACL, pages 85–90, Beijing, China.

Gustavo H. Paetzold and Lucia Specia. 2016. Unsupervised lexical simplification for non-native speakers. In Dale Schuurmans and Michael P. Wellman, editors, *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, pages 3761–3767. AAAI Press.

H. Ringbom. 1986. Crosslinguistic influence and the foreign language learning process. In E. Kellerman and Smith Sharwood M., editors, *Crosslinguistic Influence in Second Language Acquisition*. Pergamon Press, New York.

Carolina Scarton and Lucia Specia. 2018. Learning simplifications for specific target audiences. In *ACL (2)*, pages 712–718. Association for Computational Linguistics.

Carolina Scarton, Alessio Palmero Aprosio, Sara Tonelli, Tamara Martín Wanton, and Lucia Specia.

2017. Musst: A multilingual syntactic simplification tool. In *Proceedings of the IJCNLP 2017, System Demonstrations*, pages 25–28. Association for Computational Linguistics.

Advaith Siddharthan. 2010. Complex lexico-syntactic reformulation of sentences using typed dependency representations. In *Proceedings of the 6th International Natural Language Generation Conference (INLG 2010)*, Dublin, Ireland.

Adam St Arnaud, David Beck, and Grzegorz Kondrak. 2017. Identifying cognate sets across dictionaries of related languages. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2519–2528, Copenhagen, Denmark, September. Association for Computational Linguistics.

Robert A Wagner and Michael J Fischer. 1974. The string-to-string correction problem. *Journal of the ACM (JACM)*, 21(1):168–173.

William E Winkler. 1990. String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage.

Xingxing Zhang and Mirella Lapata. 2017. Sentence simplification with deep reinforcement learning. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel, editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 584–594. Association for Computational Linguistics.