CENTAL at TSAR-2022 Shared Task: How Does Context Impact BERT-Generated Substitutions for Lexical Simplification?

Rodrigo Wilkens[1], **David Alfter**, **Rémi Cardon**[1], **Isabelle Gribomont**[1,2], **Adrien Bibal**[1], **Patrick Watrin**[1], **Marie-Catherine de Marneffe**[1], **Thomas François**[1]
[1]CENTAL, IL&C, University of Louvain, Belgium
[2]Royal Library of Belgium (KBR)
{first name dot last name}@uclouvain.be

Abstract

Lexical simplification is the task of substituting a difficult word with a simpler equivalent for a target audience. This is currently commonly done by modeling lexical complexity on a continuous scale to identify simpler alternatives to difficult words. In the TSAR shared task, the organizers call for systems capable of generating substitutions in a zero-shot-task context, for English, Spanish and Portuguese. In this paper, we present the solution we (the CENTAL team) proposed for the task. We explore the ability of BERT-like models to generate substitution words by masking the difficult word. To do so, we investigate various context enhancement strategies, that we combined into an ensemble method. We also explore different substitution ranking methods. We report on a post-submission analysis of the results and present our insights for potential improvements. The code for all our experiments is available at https://gitlab.com/Cental-FR/cental-tsar2022.

1 Introduction

Lexical Simplification (LS) aims at identifying words that are considered too difficult for a given audience and replacing them with simpler substitutes.[1] Following Housen and Simoens (2016, 166), we distinguish the notion of *absolute complexity* that refers to "inherent linguistic properties of a language feature" from the notion of *difficulty*, which depends on "how costly, demanding, or difficult a given language feature is for a given language learner in a given learning context, particularly in terms of the mental resources allocated and cognitive mechanisms."

The TSAR shared task (Saggion et al., 2022) asks for solutions generating and ranking substitutes for predefined difficult words in sentences in English, Spanish and Portuguese. This paper describes the CENTAL team solution to the TSAR shared task, which takes advantage of pretrained neural language models and is easy to use in any language for which such models exist. Our solution has two steps: Substitution Generation (SG) and Substitution Ranking (SR). For SG, we use an ensemble of BERT-like models to generate candidate words to replace the difficult word. We assume language models can produce correct substitutes but are noisy (i.e., they also produce wrong substitutes). We try to mitigate this issue by combining the output of different language models in an SR step. We explore three strategies for combining and ranking the output of our SG methods. We propose a simple voting strategy for the substitutions generated by each model. We also use a standard ranking method, assuming that the ensemble of models can generate relevant substitution words, but the models do not agree on them. The third strategy uses a model trained for one language and ranks in the other two. It assumes we have poor resources for a given language and explores the use of cross-lingual transfer learning.

The remainder of this paper is organized as follows: Section 2 describes the task proposed in the TSAR shared task, their corpora and the additional corpora that we use. Section 3 details the proposed solution for generating and ranking substitutions while their results are shown in Section 4. Finally, in Section 5, we present the error analysis and possible solutions for improving the performance of the proposed methods.

2 Task and Corpora

The TSAR shared task proposes a zero-shot task, where a trial set composed of only 10 trial sentences with difficult words and their substitutions and later assessed the systems on a test corpus for English, Spanish and Portuguese. The corpus consists of sentences with one difficult word per sentence to be substituted. The TSAR corpus is consti-

[1]For a recent description of Text Simplification and Lexical Complexity, see North et al. (2022).

Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022), pages 231 - 238

tuted of 1,115 sentences with target words (373 for English, 368 for Spanish and 374 for Portuguese) annotated by 25 crowdsourced workers, whose sociodemographics are not provided. They proposed simpler substitutions for the difficult words, taking the sentence as context. An expert later selected the proposals and only non-multiword expressions were kept (Saggion et al., 2022).[2]

We used additional corpora for parameter optimization and hyperparameter tuning of the classification algorithm used in our ranking approach, given the zero-shot nature of the task. For English, we used a monolingual lexical simplification corpus (Specia et al., 2012) constituted of 2,010 English sentences annotated with difficult words and their ranked substitute words or phrases. For Spanish, we selected a cross-lingual lexical substitution corpus (Mihalcea et al., 2010) constituted of 1,300 English sentences, which are a subset of the monolingual corpus, in which the substitutes are in Spanish. To obtain both sentences and substitutions in Spanish and Portuguese, we used the Google Vision Translation API to translate the English sentences from the cross-lingual corpus to Spanish and the sentences and substitutions from the monolingual corpus to Portuguese. After translating the corpora, we automatically marked the difficult words using the list of substitutions (i.e., simpler words).[3] We divided this corpus into 80% for training and hyperparameter tuning (using cross-validation) and 20% for testing. The testing part is used for internal comparison of the methods described in Sections 3.1 and 3.2 and the training part is used in the ranking method (Section 3.2).

3 Our Approach

We detail here the runs submitted (2 for English and 3 for Spanish and Portuguese each). Figure 1 illustrates our pipeline, and Table 6, in Appendix A, shows outputs of the different strategies.

3.1 Substitution Generation

For this step, we explored whether masked BERT as a word-level "generative" model – i.e., pretrained BERT – is able to produce a suitable list of substitution candidates. Simply masking the difficult word gave unsatisfactory results in our preliminary tests. We thus investigated different ways of providing context to help the model generate adequate substitutions. All runs had words proposed by a BERT-like model, which was fed the original sentence with a mask replacing the difficult word, preceded by more context. We truncated the number of contexts generated when the concatenation of the context and the original sentence is longer than BERT models' input size limit (512 tokens). To generate that context, we explored three strategies: *Copy*, *Query Expansion*, and *Paraphrase*.

The *Copy* strategy is inspired by LSBERT (Qiang et al., 2021). The extra context preceding the sentence is simply a copy of the sentence itself. In this approach, we tested using the [SEP] token for splitting the sentences, but our experiments showed that using it led to worse results.

The *Query Expansion* (QE) strategy consists in applying the technique with the same name from the Information Retrieval domain. In our case, we produced 5 related words for the difficult word using FastText models in addition to the original sentence. We explored two variations: (1) repeating the entire sentence for each alternative, using the generated word instead of the original word, and (2) only using the proposed words.

The *Paraphrase* strategy generates a context composed of paraphrases of the original sentence. We generated up to 10 paraphrases for each sentence. The number of paraphrases is limited so that the entire prompt fits within the limit of 512 tokens imposed by BERT. This method was only applied to the English part of the shared task because, to our knowledge, there is no equivalent of the applied model for Portuguese and Spanish.

In our experiments, we compared various models available on HuggingFace[4] and observed different behaviors depending on the strategy.[5] For the official submission, we chose those that produced the best results on the test corpus. Thus, we combined the Large and Base models in the QE strategy and employed only Large models in the Copy strategy,[6]

[2]The original sentences came from three different datasets: the PorSimplesSent dataset for Portuguese (Leal et al., 2018) and the CWI Shared Task 2018 dataset for Spanish and English (Yimam et al., 2017).

[3]The sentences in which the difficult word could no longer be isolated in translation were dropped.

[4]https://huggingface.co/

[5]We tested the following models in addition to those we submitted: bert-base-multilingual-cased, skimai/spanberta-base-cased, PlanTL-GOB-ES/roberta-base-bne, josu/roberta-pt-br and rdenadai/BR_BERTo.

[6]The Large and Base models used are bert-large-uncased, bert-base-uncased, roberta-large and roberta-base for English, dccuchile/bert-base-spanish-wwm-cased and dccuchile/bert-base-spanish-wwm-uncased for Spanish, and neuralmind_bert-large-portuguese-cased and neuralmind_bert-base-portuguese-

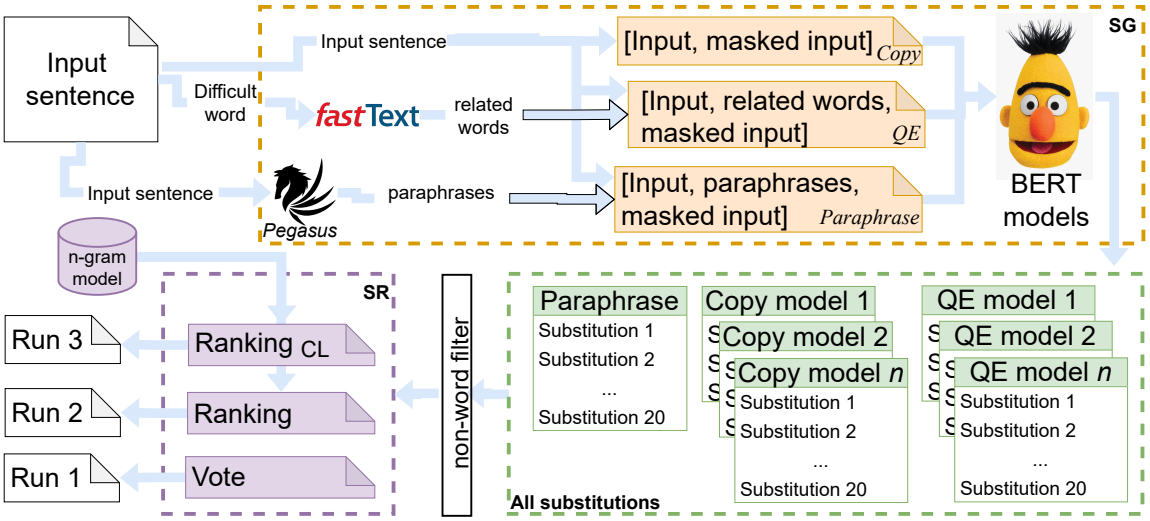

Figure 1: Pipeline of the proposed solutions.

while we used a specialized model[7] (i.e., (Zhang et al., 2020)) for the Paraphrase strategy. For the Paraphrase strategy, we used 10 beams: as we generated up to 10 paraphrases, the number of beams cannot be below 10, and there was not much difference between 10 beams and more.

The three substitution generation strategies yielded 20 items for each model. Predictions that contained non-alphabetic characters (e.g., BERT subtokens) were automatically discarded.

3.2 Substitution Ranking

All substitutions generated by the substitution generation strategies must be grouped into a single sorted list of 10 words, following the shared task guidelines. We thus combined and ranked the substitutions, selecting the top 10.

The first ranking method is a simple vote (*Vote*): we count the number of methods that generated a given substitution and rank them from most frequently proposed to least frequently proposed. This method is exemplified in Table 6.

The two other ranking methods we explored use a model of lexical complexity. High word frequency is a generally good predictor of simplicity (Brysbaert et al., 2018). However, frequencies from corpus- and list-based lookups suffer from the *out-of-vocabulary* (OOV) problem; instead, we use character-based n-gram language models to represent words (Wieting et al., 2016; Bojanowski et al., 2017). For each language, we create a character-based n-gram language model with $1 \leq n \leq 4$. The English model was trained on the *British National Corpus* (BNC Consortium, 2007). The Spanish model was trained on *Corpus lingüístico de referencia de la lengua española en Chile* (Marcos Marín, 1991). The Portuguese model was trained on *PorPopular* (Silva, 2010). We use the probabilities of each n-gram model to represent words as input for the model.

In the second ranking method (*Ranking*), we train a binary classifier on the SemEval train corpus (Section 2) predicting which one of two words is easier. For training, we concatenate the vector representations (n-gram probabilities) of two words. We opted for XGBoost (Chen and Guestrin, 2016) – with hyperparameter tuning – as the classification algorithm. We tested RandomForest, ExtraTrees, MLP, DecisionTree, AdaBoost, and Bagging classifier, all from the scikit-learn package (Pedregosa et al., 2011), including hyperparameter tuning, and found that XGBoost outperformed the other algorithms. It calculates scores based on pairwise comparisons between words and produces a ranking over a list of substitution words.

As a third ranking method (*Ranking*$_{CL}$), we explore the cross-linguistic applicability of the English classifier model. In this setup, Spanish and Portuguese words are vectorized by their respective language model (similarly to the *monolingual ranking* method), but the ranking is performed by the English ranking model.

For all rankings, if the difficult word itself is found within the final list of ten substitutions, the list is truncated up to the difficult word, otherwise we take the top 10 substitutions. In a ranking in-

cased for Portuguese.

[7] google/pegasus-xsum model

cluding the difficult word, all words ranked after the difficult word are considered more difficult than the original difficult word itself, and are thus not good substitutions for simplification.

4 Evaluation

Our evaluation of the 8 runs submitted (one for each ranking method[8]) focuses on the MAP/Potential@1 metric (@1 in our tables). All official metrics adopted by the shared task are in Appendix A.

Lang	Method	@1	Rank
EN	$QE_{BERT_L\ 1}$	.4155	21
	$QE_{BERT_L\ 2}$	**.5281**	8
ES	$QE_{RoBERTa_L\ 1}$	.3109	10
	$QE_{RoBERTa_L\ 2}$	**.4477**	1
PT	$QE_{BERT\ 1}$	.4090	5
	$QE_{BERT\ 2}$	**.4759**	2
EN	$Copy_{B\ U}$	.4959	12
	$Copy_{L\ U}$	.5040	11
	$Copy_{RoBERTa_B}$	.4772	14
	$Copy_{RoBERTa_L}$	.3994	23
ES	$Copy_C$	.4211	2
	$Copy_U$	.2989	12
PT	$Copy_B$	.4331	4
	$Copy_L$	.4705	3
EN	Paraphrase	.2171	36

Table 1: MAP/Potential@1 of our substitution generation techniques (U: uncased, C: cased, B: base, L: large; 1 and 2 refer to the first and second variations of QE)

Table 1 shows the MAP/Potential@1 of each substitution generation strategy. *Paraphrase* gives the worst result. This method did not provide many correct substitutions (see the potential in Appendix A, Table 4). Still, the proportion between the scores is similar to the other prompt-based methods (i.e., the value of potential is about twice as high as other metrics). Overall *QE* achieved better results than *Copy*. In addition, only using the words from FastText (ignoring the sentence) as additional context (i.e., variant 2) outperforms the use of the entire sentence. In general, large (L) models tend to outperform base (B) models.[9] For the three languages, *QE* achieved the best results in terms of @1 and MAP scoring methods. It also reached the best potential for Spanish and Portuguese.

Table 2 shows the results of each run. Interestingly, *Vote* tends to provide the best results for Spanish and Portuguese. It implies that the models tend to propose the correct words. For English, the ranking method achieved the best results. It is likely due to a strong disagreement between the models for this language.

Lang	Method	@1	Rank
EN	Vote	.2761	28
	Ranking	**.3619**	23
ES	Vote	**.3097**	8
	Ranking	.1983	17
	$Ranking_{CL}$	.2201	14
PT	Vote	**.3689**	2
	Ranking	.2058	15
	$Ranking_{CL}$	.2245	10

Table 2: Official results

5 Error analysis

To better understand our results, we evaluated the substitution generation and the ranking strategies. We also measure the gap between our best ranking model (*Vote*) and a perfect substitution generation step (i.e., an oracle).

For the SG step, our methods rely on providing BERT models with a single mask, but they cannot produce multiword expressions. To identify the impact of this limitation, we calculated their proportion in the gold standard: 3.35% for English, 6.27% for Spanish, and 2.97% for Portuguese.

We also studied the extent to which substitutions generated by our methods were grammatically correct regarding the context. To do so, we compared the morpho-syntactic information of each candidate against its respective difficult word, after analyzing the sentences with Stanza (Qi et al., 2020), assuming the parser output is correct. Out of all the candidates present in our submitted runs, there was a mismatch in 10.68% of the cases for English, 6.09% for Spanish, and 12.28% for Portuguese. We corrected those mismatches by using DELA dictionaries (Courtois, 1990).[10] Whenever

[8] The cross-lingual ranking method is not used for English because we only use this language as a pivot.

[9] The superior performance of the large models is in line with our experiments. However, we note that we identify exceptions such as $Repeat_{B\ U}$ for English.

[10] https://github.com/UnitexGramLab/

a mismatch was detected, we converted the Stanza information to the DELA format. Using the candidate's lemma, we checked whether an inflected form with the same morpho-syntactic information existed. If it did, we replaced the candidate with the correct form, otherwise, we deleted the candidate from the list. We can see that there is a slight improvement (up to .03 on MAP/Potential@1), indicating that while it solves issues, inflection is not the main shortcoming of the submitted lists.[11] In future work, we would like to apply this correction phase to each individual model's output in order to apply the ranking to morpho-syntactically correct candidates. In Table 6, the impact of the parser combined with the dictionary-based correction is illustrated in the line "POS filtered out", which indicates the percentage of reduction in the number of responses.

As for the ranking methods, we see that for Spanish and Portuguese, voting produces better results than ranking, while for English, ranking produces better results than voting. We hypothesize that voting prioritizes frequent and contextually suitable words that are generated by multiple methods, while ranking performs better on the tail end of the distribution. To test it, we used the ranking system exclusively to break ties created by the vote. This produces slightly better results than a full ranking in all cases, indicating that the ranking does indeed learn about simple words, yet does not have enough information on its own to rank a full list in the order given by the gold standard.

We also explored the importance of a substitution selection method, instead of a simple filter. To do so, we analyze the best possible results using all the generated substitutions for the voting method. So, we drop all generated words that are not in the gold standard and apply the same voting method. This substitution selection is exemplified in the line "Oracle+SS" in Table 6. This showed a considerable increase in voting performance (a gain of 0.7212 for English, 0.5544 for Spanish and 0.5268 for Portuguese).[12] This improvement points out the need for substitution selection methods and improvement of the ranking.

It is interesting to note that the results of the substitution generation methods outperform our ranking methods, including *Vote*, which only counts the agreement between the models. However, the previous analysis showed that the different strategies produce the correct words. This apparent contradiction is mostly due to the fact that the models can individually predict some of the correct words, but they also predict several unrelated words at the same time. Moreover, the proposed strategies share common key elements (e.g., the BERT-like model), and the *Copy* strategy, our worst result, is also present in the other two strategies. Therefore, the models' ensemble, despite agreeing on the correct words, also agree on the incorrect words. This effect is illustrated in the line "Oracle SS step filter", which indicates the percentage of removed words when applying the oracle substitution selection.

6 Conclusion

This paper presented the solution proposed by the CENTAL team in the TSAR shared task on lexical simplification. We proposed three substitution generation strategies, where we saw that Query Expansion is superior. Moreover, generation strategies can produce and sort suggestions with good performance. The Query Expansion strategy could achieve 8th, 1st and 2nd positions for English, Spanish and Portuguese respectively by itself. We also identified that the voting method might produce promising results, but a good substitution selection step is required. This step would improve morphologically incorrect substitutions and remove semantically/contextually inappropriate substitutions. In addition, the ranking methods can be useful for breaking ties in voting.[13]

Acknowledgements

We would like to thank the anonymous reviewers for their comments that helped improve the presentation of our approach and results. Rodrigo Wilkens is supported by a research convention with France Education International (FEI). David Alfter is supported by the Fonds de la Recherche Scientifique de Belgique (F.R.S-FNRS) under grant MIS/PGY F.4518.21. Rémi Cardon is supported by the FSR Incoming Postdoc Fellowship program of the FSR - Université catholique de Louvain. Isabelle Gribomont is supported by the FED-tWIN program from BELSPO. Adrien Bibal is supported by the Walloon region with a Win2Wal funding.

unitex-lingua/tree/master/

[11] Table 5 shows the results obtained for our submitted runs after applying this method.

[12] See Table 4 for all metrics.

[13] The code for our models is available at https://gitlab.com/Cental-FR/cental-tsar2022.

Marie-Catherine de Marneffe is a Research Associate of the Fonds de la Recherche Scientifique – FNRS.

References

BNC Consortium. 2007. British National Corpus. *Oxford Text Archive Core Collection*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Marc Brysbaert, Paweł Mandera, and Emmanuel Keuleers. 2018. The word frequency effect in word processing: An updated review. *Current Directions in Psychological Science*, 27(1):45–50.

Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA. ACM.

Blandine Courtois. 1990. Un système de dictionnaires électroniques pour les mots simples du français. *Langue Francaise*, 87:11–22.

Alex Housen and Hannelore Simoens. 2016. Introduction: Cognitive perspectives on difficulty and complexity in l2 acquisition. *Studies in Second Language Acquisition*, 38(2):163–175.

Sidney Evaldo Leal, Magali Sanches Duran, and Sandra Aluísio. 2018. A nontrivial sentence corpus for the task of sentence readability assessment in portuguese. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 401–413.

Francisco Marcos Marín. 1991. Corpus lingüístico de referencia de la lengua española. *Boletín de la Academia Argentina de Letras*, 56(1991):129–155.

Rada Mihalcea, Ravi Sinha, and Diana McCarthy. 2010. SemEval-2010 task 2: Cross-lingual lexical substitution. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 9–14, Uppsala, Sweden. Association for Computational Linguistics.

Kai North, Marcos Zampieri, and Matthew Shardlow. 2022. Lexical complexity prediction: An overview. *ACM Computing Surveys (CSUR)*.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *the Journal of machine Learning research*, 12:2825–2830.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. *arXiv preprint arXiv:2003.07082*.

Jipeng Qiang, Yun Li, Yi Zhu, Yunhao Yuan, Yang Shi, and Xindong Wu. 2021. Lsbert: Lexical simplification based on bert. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3064–3076.

Horacio Saggion, Sanja Štajner, Daniel Ferrés, Kim Cheng Sheang, Matthew Shardlow, Kai North, and Marcos Zampieri. 2022. Findings of the TSAR-2022 Shared Task on Multilingual Lexical Simplification. In *Proceedings of TSAR workshop held in conjunction with EMNLP 2022*.

Bruna Rodrigues da Silva. 2010. PorPopular: o português popular escrito em um objeto de aprendizagem.

Lucia Specia, Sujay Kumar Jauhar, and Rada Mihalcea. 2012. SemEval-2012 task 1: English lexical simplification. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 347–355, Montréal, Canada. Association for Computational Linguistics.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2016. Charagram: Embedding words and sentences via character n-grams. *arXiv preprint arXiv:1607.02789*.

Seid Muhie Yimam, Sanja Štajner, Martin Riedl, and Chris Biemann. 2017. Cwig3g2-complex word identification task across three text genres and two user groups. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 401–407.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

A Appendix

The Appendix presents a complete version of the results, which have been shortened in the main text due to space constraints. Table 3 shows the results from the Substitution Generation strategies discussed in Section 3. Table 4 shows the results of the submitted runs, as presented in Section 4, as well as the value we would obtain with a perfect substitution filtering step before ranking. This upper bound is calculated by removing all words that are not in the gold standard before the ranking. Table 5 shows the results after automatically correcting the results presented in Table 4. In these tables, the best results of each language are in bold (the statistical significance is not calculated). We also indicate the rank of each method (based on the @1 column) in comparison with the official results. Moreover, the MAP/Potential@1 is titled "@1".

In addition, Table 6 presents some examples of outputs of the different strategies and the results of the voting method presented in Section 3. It also illustrates the impact of the substitution selection method discussed in Sections 4 and 5.

				MAP			Potential			Accuracy		
Lang	**Method**	**@1**	@3	@5	@10	@3	@5	@10	@1	@2	@3	**Rank**
EN	$QE_{Bert_L\ 1}$	.4155	.2752	.2142	.1365	.7050	.7855	.8873	.1903	.3029	.3753	21
	$QE_{Bert_L\ 2}$	**.5281**	**.3431**	**.2554**	**.1640**	.7640	.8659	.9195	**.2627**	**.3914**	.4611	8
ES	$QE_{Roberta_L\ 1}$	.3109	.2397	.1867	.1208	.5898	.7345	.8632	.1179	.2091	.2868	10
	$QE_{Roberta_L\ 2}$	**.4477**	**.2983**	**.2222**	**.1410**	**.7265**	**.8364**	**.9383**	**.2037**	**.3029**	**.3860**	1
PT	$QE_{Bert\ 1}$	.4090	.2473	.1794	.1041	.6577	.7433	.8101	.2112	.3235	.3716	5
	$QE_{Bert\ 2}$	**.4759**	**.2892**	**.2055**	**.1189**	**.7139**	.7727	**.8422**	**.2540**	.3609	.4090	2
EN	$Repeat_{B\ U}$	.4959	.3296	.2496	.1587	.7479	.8525	.9276	**.2627**	.3833	.4611	12
	$Repeat_{L\ U}$	.5040	.3245	.2466	.1579	.7506	.8552	.9302	.2520	.3619	.4450	11
	$Repeat_{Robert_B}$	.4772	.3263	.2497	.1604	**.7962**	**.8793**	**.9490**	.2359	.3753	**.4745**	14
	$Repeat_{Roberta_L}$	.3994	.2634	.1996	.1216	.7131	.8069	.8981	.1581	.2654	.3565	23
ES	$Repeat_C$	.4211	.2601	.1952	.1111	.6467	.7255	.7880	.1956	.2744	.3396	2
	$Repeat_U$	.2989	.1840	.1298	.0744	.4809	.5489	.6250	.1413	.2092	.2364	12
PT	$Repeat_B$	.4331	.2693	.1985	.1176	.6925	.7513	.8208	.2513	.3342	.3957	4
	$Repeat_L$	.4705	.2843	.1984	.1158	.7032	**.7807**	.8395	.2513	**.3689**	**.4144**	3
EN	Paraphrase	.2171	.1407	.1069	.0650	.3833	.4638	.5603	.0938	.1581	.1849	36

Table 3: Results of each candidate generation strategy. @1 indicates the MAP/Potential@1 (U: uncased, C: cased, B: base, L: large; 1 and 2 refer to the first and second variations of QE)

				MAP			Potential			Accuracy		
Lang	**Method**	**@1**	@3	@5	@10	@3	@5	@10	@1	@2	@3	**Rank**
EN	SS+Vote	.9973	.9678	.8643	.5182	.9973	.9973	.9973	.3833	.6219	.7372	1
	Vote	.2761	.1635	.1183	.0707	.3780	.4021	.4182	**.1313**	.1930	.2117	29
	Ranking	**.3619**	**.2573**	**.2056**	**.1271**	**.6541**	**.7667**	**.8418**	.1152	**.2091**	**.2788**	24
ES	SS+Vote	.8641	.7083	.5103	.2649	.8641	.8641	.8641	.9097	.4211	.5244	1
	Vote	**.3097**	**.1826**	**.1327**	**.0779**	**.5000**	.5923	.6358	**.1467**	**.2092**	**.2391**	9
	Ranking	.1983	.1265	.0979	.0695	.4184	.5570	.7282	.0652	.1114	.1657	18
	$Ranking_{CL}$	.2201	.1416	.1122	.0745	.4646	**.6086**	**.7581**	.0407	.0896	.1331	15
PT	SS+Vote	.8957	.7103	.5235	.2737	.8957	.8957	.8957	.3101	.4786	.5401	1
	Vote	**.3689**	**.1983**	**.1344**	.0766	**.5240**	.5641	.6096	**.1737**	**.2433**	**.2673**	3
	Ranking	.2058	.1470	.1103	.0726	.4786	.6016	.7673	.0641	.1203	.1898	16
	$Ranking_{CL}$	.2245	.1478	.1143	**.0769**	.4705	**.6096**	**.8021**	.0614	.1310	.1925	11

Table 4: Results of the candidate ranking strategies. @1 indicates the MAP/Potential@1 Official results (CL: cross-language). SS+Vote refers to the study of an oracle substitution selection combined with voting.

Lang	Method	MAP @1	MAP @3	MAP @5	MAP @10	Potential @3	Potential @5	Potential @10	Accuracy @1	Accuracy @2	Accuracy @3
EN	Vote	.2815	.165	.1204	.0708	.3753	.3994	.4128	.1367	.193	.2117
	Ranking	.3646	.2622	.2084	.1267	.6541	.764	.8257	.1152	.2091	.2815
ES	Vote	.3179	.1911	.1389	.0815	.5135	.6086	.6603	.1467	.2119	.25
	Ranking	.2201	.1394	.1061	.0741	.451	.5788	.7527	.076	.1222	.182
	$\text{Ranking}_{\text{CL}}$	.2282	.1493	.118	.078	.4864	.6304	.7826	.0489	.1005	.1467
PT	Vote	.3877	.2039	.1401	.0792	.5427	.5775	.6229	.1818	.254	.2754
	Ranking	.2192	.1552	.1175	.0758	.4946	.6256	.7807	.0721	.1336	.2058
	$\text{Ranking}_{\text{CL}}$	.2326	.1555	.1206	.0799	.4973	.6417	.8155	.0721	.147	.2112

Table 5: Results obtained by applying the DELA correction method to the submitted runs (Table 2). @1 indicates the MAP/Potential@1 (CL: cross-language).

	Lebanon is sharply split along **sectarian** lines, with 18 religious sects.	The **motive** for the killings was not known.
$\text{QE BERT}_{\text{L 1}}$	religious sunni secular islamist islamic [...] shia	motive reason motivation motives purpose [...] impetus
$\text{QE BERT}_{\text{L 2}}$	religious ideological ethnic regional national [...] tribal	motive reason motivation motives purpose [...] plan
Copy_{U}	religious ethnic secular national islamic [...] shia	motive reason motivation cause motives [...] blame
$\text{Copy}_{\text{RoBERTa L}}$	sectarian religious theological spiritual sunni [...] dramatic	motive reason rationale motives cause [...] target
Paraphrase	religious ethnic ideological cultural religion [...] many	the punishment location a information [...] reasons
Non-word filtered	-	. " (
Vote	religious (5) secular (5) ethnic (4) regional (4) protestant (4)	motive (5) reason (5) motivation (4) motives (4) purpose (4)
POS filtered out	80% of words removed	98% of words removed
Vote after POS filter	religious (5) secular (5) ethnic (4) regional (4) political (4)	reason (5) motive (4) cause (4) intention (2) inspiration (1)
Oracle SS step filter	92% of words removed	90% of words removed
Oracle SS+Vote	religious (5) sectarian (1) provincial (1) party (1)	criminals (4)

Table 6: Outputs of the substitution generation (SG) methods and the Vote ranking strategy (Section 3) for two examples, as well as the evaluation and analysis performed in Sections 4 and 5. We give the top 5 candidates and the last for each SG method.