LyS_ACoruña at SemEval-2022 Task 10: Repurposing Off-the-Shelf Tools for Sentiment Analysis as Semantic Dependency Parsing

Iago Alonso-Alonso, David Vilares and Carlos Gómez-Rodríguez
Universidade da Coruña, CITIC
Departamento de Ciencias de la Computación y Tecnologías de la Información
Campus de Elviña s/n, 15071
A Coruña, Spain
{iago.alonso,david.vilares,carlos.gomez}@udc.es

Abstract

This paper addressed the problem of structured sentiment analysis using a bi-affine semantic dependency parser, large pre-trained language models, and publicly available translation models. For the monolingual setup, we considered: (i) training on a single treebank, and (ii) relaxing the setup by training on treebanks coming from different languages that can be adequately processed by cross-lingual language models. For the zero-shot setup and a given target treebank, we relied on: (i) a word-level translation of available treebanks in other languages to get noisy, unlikely-grammatical, but annotated data (we release as much of it as licenses allow), and (ii) merging those translated treebanks to obtain training data. In the post-evaluation phase, we also trained cross-lingual models that simply merged all the English treebanks and did not use word-level translations, and yet obtained better results. According to the official results, we ranked 8th and 9th in the monolingual and cross-lingual setups.

1 Introduction

Sentiment Analysis (SA, Pang and Lee, 2008) deals with the automatic processing of subjective information in natural language texts. Early work on SA focused on conceptually simpler tasks, such as polarity classification at the sentence or document level. With the advances in natural language processing (NLP), more fine-grained and complex tasks have been proposed, such as detecting the entity that expresses an opinionated chunk of text, or the entity that was targeted. More particularly, Barnes et al. (2021) consider sentiment analysis as a (graph) structured task, and discuss up to five subtasks: (i) sentiment expression extraction, (ii) sentiment target extraction, (iii) sentiment holder extraction, (iv) defining the relationship between these elements, and (v) assigning a polarity label. They discuss that although these tasks have been extensively studied by different authors (Turney, 2002; Pontiki et al., 2015; Zhang et al., 2019, *inter alia*), they are not addressed all together. They also discuss that such subdivision into subtasks might have a negative impact in the general analysis of the sentence, and that a joint analysis could translate into a holistic approach. To do so, they propose to encapsulate all these tasks in the form of a sentiment graph. Formally, the goal is to find the set of opinion tuples $\{O_1, \ldots, O_i, \ldots, O_n\}$ in a given text, where each opinion O_i is a tuple of the form (h, t, e, p) where h is a holder who expresses a polarity p towards a target t through a sentiment expression e, implicitly defining pairwise relationships between elements of the same tuple. We illustrate an example in Figure 1.

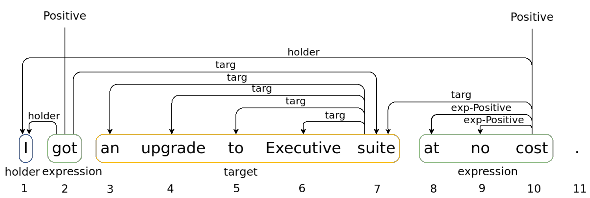

Figure 1: An example of sentiment graph as defined by Barnes et al. (2021). The sentence has a holder ('I'), two sentiment expressions ('got' and 'at no cost') and one target ('an upgrade to Executive suite')

More particularly, for the SemEval-2022 Task 10 (Barnes et al., 2022), the organizers proposed both a monolingual[1] and a cross-lingual (zero-shot) setup. They considered 5 languages (and 7 treebanks): English, Spanish, Catalan, Basque, and Norwegian. For the zero-shot setup Basque, Catalan, and Spanish were the target languages.

Our approach is based on the idea of viewing this task as semantic dependency parsing (Oepen et al., 2015), since both tasks are structurally similar even if the graphs have different meaning. More

[1]We use the term monolingual as it was the term used by the organizers, but this setup allowed the use of any resource, including resources in different languages.

Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), pages 1389 - 1400
July 14-15, 2022 ©2022 Association for Computational Linguistics

specifically, we rely on a bi-affine graph-based parser (Dozat and Manning, 2018) and different large pre-trained language models (LM), such as BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019) or XLM-R (Conneau et al., 2020). For the monolingual setup we train a semantic parsing model on single and merged treebanks, and compare the performance using different LMs. For the cross-lingual setup, we first do a word-level translation of the datasets in a different language than the target treebank, and then proceed similarly to the monolingual setup. Overall, the approach relies on off-the-shelf tools already available, but traditionally used for other purposes. We here re-purpose them for their use for sentiment analysis as graph-based parsing.

2 The role of parsing in SA

Parsing has been used in the past for SA, with different motivations, such as integrating syntactic knowledge as a component of the model's architecture, or producing structured sentiment outputs.

Polarity classification. Since early times, authors have studied the importance of language structure to deal with relevant linguistic phenomena for polarity classification, first focusing on simpler strategies such as the use of n-grams or lexical rules (Pang et al., 2002; Taboada et al., 2011). Later on, more complex syntactic structures were incorporated as well, both for rule-based and machine learning approaches.

For instance, for the rule-based paradigm, Poria et al. (2014) used dependency relations for concept-level sentiment analysis, so sentiment could flow from one concept to another to better contextualize polarity. Vilares et al. (2015a, 2017) proposed a model to compute the sentiment of sentences that was driven by syntax-based rules to deal with specific relevant phenomena in SA, and that could be easily re-purposed for any language for which a dependency parser was available. Kanayama and Iwamoto (2020) built on top of Vilares et al.'s idea, and proposed a multilingual syntax-based system that achieved a high precision for 17 languages.

From the machine learning perspective, Joshi and Rosé (2009); Vilares et al. (2015b) used dependency triplets to train data-driven (pre-neural) models and obtain slight improvements over purely lexical approaches. Socher et al. (2013) collected sentiment labels for phrases and sentences that were previously automatically represented as constituent (sub)trees, to then train a compositional model that used a recursive neural network. This work has some relevant resemblances with Barnes et al. (2021)'s proposal for structured sentiment analysis. Socher et al. were among the first to provide tree-shaped annotated sentiment data (in this case just for polarity classification), while most of previous work had focused on using tree knowledge as external information to the models, but with sentiment annotations only associated with plain texts. This publicly available data later encouraged many authors to design models that could exploit tree-shaped annotated data to obtain better performing models (Tai et al., 2015; Zhang and Zhang, 2019, *inter alia*).

Aspect-based sentiment analysis (ABSA). ABSA is a task that is particularly suitable for the integration of syntactic information, since its main goal is to associate sentiment with specific entities and aspects that occur in the sentence (Pontiki et al., 2015). Related to this, Popescu and Etzioni (2005) already used dependency trees to constrain an unsupervised sentiment analysis system that extracted a set of product features and their sentiment, given a particular item. More recently, with the wide adoption of neural networks in NLP, different authors have integrated syntactic knowledge and syntactic structures in different network architectures, such as long short-term memory networks (LSTMs, Tang et al., 2016), recursive neural networks (Nguyen and Shirai, 2015), convolutional networks (Xue and Li, 2018), and graph attention networks (Huang et al., 2020; Sun et al., 2019).

Overall, it is clear that parsing has had a high relevance in SA. Yet, the novelty of the shared task is in using graphs to represent richer annotations. This makes it possible to use parsing algorithms as sentiment models, i.e. not just to use them as a component of the model architecture, but as the model responsible of producing the whole sentiment structure of the chunk of text. Also, this is especially relevant in the era of large neural models, where the utility of parsers for downstream tasks is sometimes questioned, with some studies questioning its need in the presence of pretrained models that implicitly learn syntax (Tenney et al., 2019; Glavaš and Vulić, 2021; Dai et al., 2021) while others still achieve extra accuracy from their use in conjuntion with such models (Sachan et al., 2021; Xu et al., 2021; Li et al., 2021; Zhang et al., 2022). In any case, tasks like this one show that graph structures can be also useful to re-purpose traditional tasks such as SA, while taking advantage of research that the NLP community has done on parsing algorithms for decades.

3 Brief overview of the shared task

The goal of the task is to produce graph structures that reflect the sentiment of a sentence, as we showed in Figure 1. More particularly, the organizers released 7 treebanks in 5 different languages: OpeNER (Agerri et al., 2013, English and Spanish), MPQA (Wiebe et al., 2005, English), Darmstadt_unis (Toprak et al., 2010, English), MultiBooked (Barnes et al., 2018, Basque and Catalan), and NoReC_fine (Øvrelid et al., 2020, Norwegian).[2] Table 1 details the main statistics for the datasets.

Dataset	Language	# sents	# holders	# targets	# expr.
NoReC_fine	Norwegian	11437	1128	8923	11115
MultiBooked	Basque	1521	296	1775	2328
MultiBooked	Catalan	1678	235	2336	2756
OpeNER	Spanish	2057	255	3980	4388
OpeNER	English	2494	413	3850	4150
MPQA	English	10048	2279	2452	2814
Darmstadt_unis	English	2803	86	1119	1119

Table 1: General statistics of the treebanks used in the shared task.

The sentiment of a sentence is composed of all the opinions, O_i, that make it up. Each opinion can have up to four elements: a holder (h) who expresses a polarity (p) towards a target (t) through a sentiment expression (e). These four elements implicitly define the pairwise relationships between the elements of a tuple. The previous example, Figure 1, shows a sentence with two sentiment expressions (*got* and *at no cost*) that express the polarity (*Positive*) of the sentiment that a holder (*I*) has towards one target (*an upgrade to Executive suite*).

Preprocessing The organizers of the shared task proposed two possible ways to address the task: as a sequence labeling or as graph-based parsing problem. As mentioned above, we opted for the latter. We use the scripts available in the official repository to transform the JSON files to the CoNLL-U based format and *vice versa*, and we applied the needed changes to make it compatible with `supar` (see §4).[3] Under the graph-based paradigm, the problem is approached as a bilexical dependency graph prediction task, with some assumptions. To convert the data, the organizers suggest two possible conversions, namely *head-first* and *head-final*. In *head-first*, it is assumed that the first token of the sentiment expression is a root node, and that the first token of each holder or target spans is the head node of such span, while the other ones are dependents. Meanwhile, in *head-final*, the final token of the holder and target spans is set as the head of the span, and the final token of the sentiment expression as a root node (Figure 1 is a head-final example). In this work, we have chosen *head-final*, which is the default option for the shared task and also delivered better results than *head-first* in the experiments carried out by Barnes et al. (2021) (see Table 3 in that paper).

Subtasks More in detail, the challenge is divided into two subtasks:

1. Monolingual setup: When training and development data is available for the same treebank/language, i.e. the goal is to train one model per treebank. It was allowed to use extra resources or tools that could boost performance, even from different languages.

2. Cross-lingual, zero-shot setup: It is assumed that there is no gold training data in the language of the target treebank. The organizers specified that it is possible to use treebanks in other languages, translation tools, and any other resources that do not include sentiment annotations in the target language.

Metrics Each subtask is evaluated independently, and the ranking metric was *sentiment graph F1* (Barnes et al., 2021), where true positives are exact matches at the graph level, weighting the overlap between the predicted and gold spans for each element, and averaged across all three spans. To compute precision, it weights the number of correctly predicted tokens divided by the total number of predicted tokens, while for recall it weights the number of correctly predicted tokens divided by the total number of gold tokens. Also, as mentioned earlier, it is possible to have tuples with empty holders and targets.

[2] For more detailed information see https://github.com/jerbarnes/semeval22_structured_sentiment

[3] https://github.com/MinionAttack/conllu-conll-tool

4 Our model

We rely on the Dozat and Manning (2018) parser, a widely used state-of-the-art model both for syntactic and semantic dependency parsing. Inspired in previous graph-based parsers (McDonald et al., 2005; Kiperwasser and Goldberg, 2016), the parser first computes contextualized representations for each word using bidirectional LSTMs (biLSTMs; Hochreiter and Schmidhuber, 1997; Schuster and Paliwal, 1997). After that, the model computes a *head* and a *dependent* representation for each term, to establish through a bi-affine attention whether an edge exists between each pair of tokens, and if so, what is the semantic relationship between them. In particular, in this paper we follow the implementation used in the `supar`[4] package, as it has been widely adopted by the community and it is available for other flavors of parsing as well, such as constituent or dependency parsing. We preferred this implementation over the graph-based baseline provided in the SemEval repository, since early experiments showed a superior performance, and it also offered a simpler integration of large language models. We left the parser hyperparameters, except the learning rate, at their default value.

Pre-trained language models For each language we looked for avaliable monolingual and multilingual pre-trained LMs at `https://huggingface.co/`. Specifically, for each language, we included:

- `Basque`: berteus-base-cased, RoBasquERTa.
- `Catalan`: julibert, roberta-base-ca, calbert-base-uncased.
- `English`: bert-base-cased, bert-base-uncased, bert-large-cased, bert-large-uncased, roberta-base, roberta-large, albert-base-v2, albert-large-v2, xlnet-base-cased, xlnet-large-cased, electra-base-discriminator, electra-large-discriminator, electra-base-generator, electra-large-generator.
- `Norwegian`: norbert, nb-bert-base, nb-bert-large, electra-base-norwegian-uncased-discriminator.
- `Spanish`: bio-bert-base-spanish-wwm-uncased, bert-base-spanish-wwm-cased, roberta-base-bne, roberta-large-bne, selectra_medium, zeroshot_selectra_medium.

With respect to the cross-lingual LMs, we considered: xlm-roberta-base and xlm-roberta-large.

4.1 Monolingual models

For this task, we use: (i) pre-trained language models, (ii) `supar`, and (iii) the official training and development files to build our models. Also note that we train end-to-end models, using words as the only input (later tokenized into subword pieces by the language models), but ignoring the part-of-speech tags and syntactic information provided in the sentiment treebanks. We did not use part-of-speech tags (or other morphosyntactic annotations) since these are not used in `supar` together with BERT encoders, and using them would require to adapt the code, which was exactly what we tried to avoid in this work.

Training procedure We fine-tuned parsing models considering for each treebank the proposed LMs, and combining them with `supar`. Since training is time-consuming, many model configurations are proposed, and the performance of `supar` is stable independently of the seed, we decided to train a single model per LM. Specifically, all models have been trained with the default seed used by `supar`, which is 1. The only parameter that was modified was the learning rate (l_r), as we observed that for some models (specially the larger language models) the fine-tuning process did not converge. We started with $5 \cdot 10^{-5}$, and did a small grid search down to $1 \cdot 10^{-6}$, where if a model still did not converge it was discarded.[5] Additionally, to train the parsing models, we considered three strategies:

1. Single monolingual training and development files: We train each model on a single treebank and validate its performance in the corresponding dev set, i.e., the standard monolingual training and development methodology.

For the best model obtained for each treebank according to strategy 1, we explored a couple of harmonized training strategies (harmonized in the sense that different treebanks follow the same annotation guidelines):

2. Merged training and development files from different treebanks: We considered to merge

[4] `https://github.com/yzhangcs/parser`

[5] For both monolingual and cross-lingual subtasks, all the selected models used a l_r of $5 \cdot 10^{-5}$. The only exception is Norwegian in the monolingual subtask, for which we used $5 \cdot 10^{-6}$.

all the available training files and all the available development files, treating them as a single dataset. Thus, we trained a single model that could predict all test files, but with the disadvantage that model selection is based on multilingual performance, which could hurt the performance in this setup.

3. Merged training files, single development file: Similar to 2, but merging only the training files. For the development phase, we proceeded as in 1 and used each dataset's dev file for model selection. The idea was to have training data that can benefit from multilingual information, but that still considers a monolingual file for a language-dependent model selection, i.e., given n treebanks, we still need to train n models, one per treebank.

We detail the experimental results for the training/development phase in §5.

4.2 Cross-lingual (zero-shot) models

In this setup, we rely on two main components: (i) available translation systems to perform word-level translations from source language to target language treebanks, and (ii) both monolingual and cross-lingual language models. Our goal with (i) is to obtain noisy, unlikely-grammatical data, but that still can provide sentiment annotations for a given target language, exploring the viability of this approach. Regarding the learning rate, we used $5 \cdot 10^{-5}$ in all cases.

Auxiliary translation models From the CoNLLU converted files[6], we translated the sentences at the word level using the Helsinki-NLP translation models[7] (Tiedemann and Thottingal, 2020) available at `huggingface`. Table 2 lists the language pairs for which we could obtain translated versions for the cross-lingual setup.

Dataset	Language	Basque	Catalan	Spanish
NoReC_fine	Norwegian			✓
MultiBooked	Basque			✓
MultiBooked	Catalan			✓
OpeNER	Spanish	✓	✓	
OpeNER	English	✓	✓	✓
MPQA	English	✓	✓	✓
Darmstadt_unis	English	✓	✓	✓

Table 2: Treebanks and the languages to which they were translated for the cross-lingual experiments.

[6]https://github.com/MinionAttack/corpus-translator

[7]https://huggingface.co/Helsinki-NLP

Then, to train the models we proceeded similarly to strategy 2 used in the monolingual setup: we combined the translated training and validation files coming from treebanks in other languages, and used the micro-averaged F1-score on the translated development set for model selection.

Post-evaluation (and better) baseline After the deadline to submit proposals, we also tested a baseline consisting on training, using an XLM-RoBERTa LM as the base component, a cross-lingual model that uses all the English datasets (without any kind of translation) as the source data. We discuss these results as well in §5.2.

5 Results

Here, we detail and discuss the results that we got for both subtasks (see §5.1 and 5.2) on: (i) the official development sets, and (ii) the official test sets of the shared tasks.

5.1 Monolingual setup

Tables 3 and 4 show the results for the development phase on the English and non-English datasets, respectively, including different LMs and training setups.

With respect to the results on the English treebanks, an interesting trend is that despite being the monolingual setup, using cross-lingual language models, and in particular XLM-RoBERTa, performed surprisingly well. Combined with the training strategy 3 (merged training sets, single development set), such models obtained the best results for 2 out of 3 English corpora (OpeNER and Darmstadt), while they still ranked well in the other dataset (MPQA). Across monolingual LMs, we also observe trends: electra-base-discriminator and (both base and large) RoBERTa models obtain overall the best results. On the other hand, we did not obtain equally robust results with ALBERT, and to a lesser extent, with BERT architectures. This is not totally surprising, since among the tested LMs, BERT is among the oldest ones, and ALBERT is a lite BERT, so some computational power is lost and it is understandable that this translates into some performance loss too, compared to larger LMs.

With respect to the experiments on the non-English datasets, we observe certain similarities, although the number of available models is much smaller than in the English cases. Again, XLM-RoBERTa overall obtains the best results. The only

Corpus	Model	Strategy	F1
OpeNER_en	xlm-roberta-large	3	0.714
	electra-base-discriminator	1	0.710
	xlm-roberta-large	1	0.707
	xlm-roberta-base	2	0.686
	roberta-large	1	0.683
	xlnet-base-cased	1	0.681
	xlm-roberta-base	3	0.679
	xlm-roberta-base	1	0.673
	roberta-base	1	0.663
	electra-large-discriminator	1	0.662
	bert-large-uncased	1	0.660
	electra-large-generator	1	0.652
	xlm-roberta-large	2	0.643
	bert-base-uncased	1	0.640
	bert-large-cased	1	0.640
	xlnet-large-cased	1	0.639
	bert-base-cased	1	0.612
	electra-base-generator	1	0.612
	albert-base-v2	1	0.590
	albert-large-v2	1	0.297
MPQA	roberta-base	1	0.374
	roberta-large	1	0.365
	electra-base-discriminator	1	0.351
	xlm-roberta-large	1	0.346
	xlnet-base-cased	1	0.338
	xlm-roberta-large	3	0.327
	bert-base-cased	1	0.306
	xlm-roberta-base	2	0.303
	electra-large-generator	1	0.301
	xlm-roberta-large	2	0.298
	bert-large-uncased	1	0.297
	bert-base-uncased	1	0.294
	xlm-roberta-base	3	0.285
	xlm-roberta-base	1	0.277
	bert-large-cased	1	0.269
	electra-base-generator	1	0.253
	albert-base-v2	1	0.236
	xlnet-large-cased	1	0.209
Darmstadt_unis	xlm-roberta-large	3	0.329
	xlm-roberta-large	1	0.309
	electra-base-discriminator	1	0.306
	xlm-roberta-base	3	0.306
	xlm-roberta-large	2	0.301
	roberta-base	1	0.301
	xlm-roberta-base	1	0.276
	xlnet-base-cased	1	0.276
	xlnet-large-cased	1	0.269
	roberta-large	1	0.268
	electra-large-generator	1	0.267
	electra-large-discriminator	1	0.264
	xlm-roberta-base	2	0.262
	bert-large-uncased	1	0.257
	bert-base-uncased	1	0.251
	bert-large-cased	1	0.237
	electra-base-generator	1	0.229
	albert-base-v2	1	0.217
	bert-base-cased	1	0.212

Table 3: Scores on the development set for the English treebanks and the monolingual setup. Models trained on the training data before its updated version.

Corpus	Model	Strategy	F1
NoReC_fine	nb-bert-large	1	0.479
	nb-bert-base	1	0.459
	xlm-roberta-large	1	0.450
	xlm-roberta-large	3	0.439
	xlm-roberta-large	2	0.427
	xlm-roberta-base	3	0.414
	xlm-roberta-base	2	0.411
	xlm-roberta-base	1	0.401
	electra-base-norwegian -uncased-discriminator	1	0.382
	norbert	1	0.298
MultiBooked_eu	xlm-roberta-large	3	0.662
	xlm-roberta-large	2	0.623
	xlm-roberta-base	3	0.613
	berteus-base-cased	1	0.602
	xlm-roberta-base	2	0.597
	xlm-roberta-base	1	0.571
	xlm-roberta-large	1	0.569
	RoBasquERTa	1	0.496
MultiBooked_ca	xlm-roberta-base	1	0.694
	xlm-roberta-large	1	0.683
	xlm-roberta-large	2	0.679
	xlm-roberta-large	3	0.679
	xlm-roberta-base	2	0.674
	roberta-base-ca	1	0.672
	xlm-roberta-base	3	0.653
	julibert	1	0.590
	calbert-base-uncased	1	0.579
OpeNER_es	xlm-roberta-large	3	0.666
	xlm-roberta-base	2	0.662
	xlm-roberta-base	3	0.657
	xlm-roberta-large	2	0.639
	xlm-roberta-base	1	0.635
	xlm-roberta-large	1	0.635
	bert-base-spanish-wwm-cased	1	0.630
	selectra_medium	1	0.622
	roberta-base-bne	1	0.616
	zeroshot_selectra_medium	1	0.610
	roberta-large-bne	1	0.605
	bio-bert-base-spanish-wwm-uncased	1	0,457

Table 4: Scores on the development set for the non-English treebanks and the monolingual setup. Models trained on the training data before its updated version.

exception is the Norwegian dataset, where we obtained the best results with a BERT architecture.

Yet, a more thoughtful discussion would be needed to determine if some architectures truly behave better than others. Note that all these LMs are usually pre-trained using different and heterogeneous text sources, and specially for the less-resourced languages, some constraints are usually imposed during training. For instance, it is hard to conclude that berteus-base-cased (BERT) (Agerri et al., 2020) is worse than XLM-RoBERTa (Conneau et al., 2020), since the amount of resources to train the former was more constrained.

Finally, a few days before the submission deadline, the training files of some treebanks were slightly updated by the organizers, due to minor bugs in the segmentation process that corrupted some sentences. As we did not have time to rerun all models and update the results, we chose to re-train only the model that obtained the best performance on the previous version of the treebanks. Therefore, all the outputs submitted for the test sets correspond to models trained on the updated, uncorrupted files. In Table 5 we compare the performance of the models trained on the updated and deprecated versions of the training files. Overall, we observed relatively small, but non-negligible differences, usually obtaining a better performance with the updated version of the treebank.

Official results on the test sets In Table 6 we show the performance on the test sets of our submitted models, i.e. those that achieved the highest score in the corresponding development phase. The performance is stable across different test sets, obtaining slightly better results for Iberian languages. For a detailed comparison against the rest of participants, we refer the users to Appendix 11 and the

Corpus	Model	Strategy	Old F1	New F1
NoReC_fine	nb-bert-large	1	0.479	0.492
MultiBooked_eu	xlm-roberta-large	3	0.662	0.648
MultiBooked_ca	xlm-roberta-base	1	0.694	0.699
OpeNER_es	xlm-roberta-large	3	0.666	0.709
OpeNER_en	xlm-roberta-large	3	0.714	0.716
MPQA	roberta-base	1	0.374	0.374
Darmstadt_unis	xlm-roberta-large	3	0.329	0.357

Table 5: Scores on the development set for the models trained on the corrupted and uncorrupted versions of the training files, on the monolingual setup. For each treebank, we only did the comparison for the best performing model, based on the performance on the corrupted version.

Dataset	Model	Strategy	Score
NoReC_fine	nb-bert-large	1	$0.462_{(10)}$
MultiBooked_eu	xlm-roberta-large	2	$0.680_{(7)}$
MultiBooked_ca	xlm-roberta-base	1	$0.653_{(8)}$
OpeNER_es	xlm-roberta-large	3	$0.692_{(6)}$
OpeNER_en	xlm-roberta-large	3	$0.698_{(9)}$
MPQA	roberta-base	1	$0.349_{(10)}$
Darmstadt_unis	xlm-roberta-large	3	$0.414_{(8)}$

Table 6: Scores of our models, for the monolingual subtask, on each test set. Our ranking on the shared task for each test set is indicated as a subscript.

official shared task paper (Barnes et al., 2022).

The datasets of the shared task belong to different domains: OpeNER and MultiBooked deal with hotel reviews, NoReC with professional reviews in multiple domains, Darmstadt_unis (the dataset for which we obtain the second lowest scores) contains English online university reviews, and MPQA (the dataset for which we obtain the lowest scores) is about news articles annotated with opinions and other private states (i.e., beliefs, emotions, sentiments, speculations, . . .). For the two lowest-scoring datasets, they have in common that they mostly contain single-opinion sentences, whereas the other datasets tend to have more variety in the number of opinions and their distribution. For instance, ∼85% and ∼74% of the training sentences of the Darmstadt_unis and MPQA datasets have only one opinion, while the next most 'single-opinion' dataset is multibooked_eu with only ∼53% of the sentences. However, we need to perform more detailed analysis as future work to extract more robust conclusions.

5.2 Cross-lingual setup

Table 7 shows the results for the development phase for the three target languages and their datasets. Again, XLM-RoBERTa models obtain overall the best performance, although in this case it is less surprising since cross-lingual LMs are expected to suit well this kind of challenges. Similar to the case of the monolingual setup, we decided to retrain the best-performing model with the updated versions of the training files. In Table 8, we show the comparison between the corrupted and uncorrupted versions of the datasets, which contrarily to the monolingual setup, often turned out into worse performing models.

Corpus	Model	F1
Basque	xlm-roberta-base	0.434
	berteus-base-cased	0.416
	RoBasquERTa	0.323
Catalan	roberta-base-ca	0.564
	xlm-roberta-base	0.519
	julibert	0.486
	calbert-base-uncased	0.385
Spanish	xlm-roberta-base	0.605
	xlm-roberta-large	0.593
	bert-base-spanish-wwm-cased	0.583
	zeroshot_selectra_medium	0.555
	selectra_medium	0.536
	roberta-base-bne	0.515
	roberta-large-bne	0.438
	bio-bert-base-spanish-wwm-uncased	0.386

Table 7: Scores on the development set for the *translated* English treebanks and the cross-lingual setup. Models trained on the training data before its updated version.

Language	Model	Old F1	New F1
Basque	berteus-base-cased	0.416	0.424
	xlm-roberta-base	0.434	0.416
Catalan	roberta-base-ca	0.564	0.572
Spanish	xlm-roberta-large	0.593	0.570
	xlm-roberta-base	0.605	0.569

Table 8: Scores on the development set for the models trained on the corrupted and uncorrupted versions of the *translated* training files, on the cross-lingual setup. For each treebank, we only did the comparison for the best performing model, based on the performance on the corrupted version.

Finally, Table 9 shows the scores for the post-evaluation baseline (model trained on the English datasets with XLM-RoBERTa) on the dev set. Very interestingly, the results show that this baseline outperformed our word-level translation approaches. We need more analysis to understand why this happens, but we hypothesize that the larger amount of English texts XLM-RoBERTa was pre-trained on could be playing an important role.

Official results on the test sets Finally, in Table 10 we show our results on test sets of the cross-lingual, zero-shot setup, for which we obtain again stable results. Appendix 12 contains the results for all participants.

Language	Model	Score
Basque	xlm-roberta-base	0.678
	xlm-roberta-large	0.677
Catalan	xlm-roberta-base	0.598
	xlm-roberta-large	0.625
Spanish	xlm-roberta-base	0.663
	xlm-roberta-large	0.638

Table 9: Scores on the development set of the trained English models (trained on MPQA, OpeNER_en and Darmstadt_unis corpora, *without* word-level translation) for the cross-lingual subtask.

Language	Model	Score
Models using word-level translation		
Basque	berteus-base-cased	$0.509_{(8)}$
Catalan	roberta-base-ca	$0.554_{(8)}$
Spanish	xlm-roberta-large	$0.570_{(7)}$
Combined English corpora without word-level translation		
Basque	xlm-roberta-base	$0.649_{(2)*}$
	xlm-roberta-large	$0.641_{(2)*}$
Catalan	xlm-roberta-base	$0.647_{(2)*}$
	xlm-roberta-large	$0.655_{(2)*}$
Spanish	xlm-roberta-base	$0.670_{(1)*}$
	xlm-roberta-large	$0.638_{(2)*}$

Table 10: Scores of our models, for the cross-lingual subtask, on each test set. Our ranking on the shared task for each test set is indicated as a subscript. * indicates the ranking that we would obtain in the shared task using the post-evaluation baseline models.

6 Conclusion

This paper describes our participation at the SemEval Shared Task 10 on structured sentiment analysis. We participated both in the monolingual and cross-lingual (zero-shot) setups. We applied a simple, but effective approach, relying on off-the-shelf tools, traditionally used for other purposes, and used them to predict sentiment graphs instead. More particularly, for the monolingual setup, we linked pre-trained language models with bi-affine graph parsing and training over single and multiple treebanks. In the zero-shot setup, we followed a similar approach, but relied on publicly available translation models to obtain training data, by applying a word-level translation of treebanks, to then train models similarly to the monolingual setup.

Acknowledgements

This work is supported by a 2020 Leonardo Grant for Researchers and Cultural Creators from the FBBVA,[8] as well as by the European Research Council (ERC), under the European Union's Horizon 2020 research and innovation programme (FASTPARSE, grant agreement No 714150). The work is also supported by ERDF/MICINN-AEI (SCANNER-UDC, PID2020-113230RB-C21), by Xunta de Galicia (ED431C 2020/11), and by Centro de Investigación de Galicia "CITIC" which is funded by Xunta de Galicia, Spain and the European Union (ERDF - Galicia 2014–2020 Program), by grant ED431G 2019/01.

[8]FBBVA accepts no responsibility for the opinions, statements and contents included in the project and/or the results thereof, which are entirely the responsibility of the authors.

References

Rodrigo Agerri, Montse Cuadros, Sean Gaines, and German Rigau. 2013. OpeNER: Open polarity enhanced named entity recognition. In *Sociedad Española para el Procesamiento del Lenguaje Natural*, volume 51, pages 215–218.

Rodrigo Agerri, Iñaki San Vicente, Jon Ander Campos, Ander Barrena, Xabier Saralegi, Aitor Soroa, and Eneko Agirre. 2020. Give your text representation models some love: the case for Basque. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4781–4788, Marseille, France. European Language Resources Association.

Jeremy Barnes, Toni Badia, and Patrik Lambert. 2018. MultiBooked: A corpus of Basque and Catalan hotel reviews annotated for aspect-level sentiment classification. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Jeremy Barnes, Robin Kurtz, Stephan Oepen, Lilja Øvrelid, and Erik Velldal. 2021. Structured sentiment analysis as dependency graph parsing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3387–3402, Online. Association for Computational Linguistics.

Jeremy Barnes, Andrey Kutuzov, Oberländer, Laura Ana Maria, Enrica Troiano, Jan Buchmann, Rodrigo Agerri, Lilja Øvrelid, Erik Velldal, and Stephan Oepen. 2022. SemEval-2022 task 10: Structured sentiment analysis. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, Seattle. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Junqi Dai, Hang Yan, Tianxiang Sun, Pengfei Liu, and Xipeng Qiu. 2021. Does syntax matter? a strong baseline for aspect-based sentiment analysis with RoBERTa. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1816–1829, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Timothy Dozat and Christopher D. Manning. 2018. Simpler but more accurate semantic dependency parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 484–490, Melbourne, Australia. Association for Computational Linguistics.

Goran Glavaš and Ivan Vulić. 2021. Is supervised syntactic parsing beneficial for language understanding tasks? an empirical investigation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3090–3104, Online. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Lianzhe Huang, Xin Sun, Sujian Li, Linhao Zhang, and Houfeng Wang. 2020. Syntax-aware graph attention network for aspect-level sentiment classification. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 799–810, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Mahesh Joshi and Carolyn Rosé. 2009. Generalizing dependency features for opinion mining. In *Proceedings of the ACL-IJCNLP 2009 conference short papers*, pages 313–316.

Hiroshi Kanayama and Ran Iwamoto. 2020. How universal are Universal Dependencies? exploiting syntax for multilingual clause-level sentiment detection. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4063–4073, Marseille, France. European Language Resources Association.

Eliyahu Kiperwasser and Yoav Goldberg. 2016. Simple and accurate dependency parsing using bidirectional LSTM feature representations. *Transactions of the Association for Computational Linguistics*, 4:313–327.

Zhongli Li, Qingyu Zhou, Chao Li, Ke Xu, and Yunbo Cao. 2021. Improving BERT with syntax-aware local attention. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 645–653, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 523–530, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Thien Hai Nguyen and Kiyoaki Shirai. 2015. PhraseRNN: Phrase recursive neural network for aspect-based sentiment analysis. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2509–2514, Lisbon, Portugal. Association for Computational Linguistics.

Stephan Oepen, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Silvie Cinková, Dan Flickinger, Jan Hajič, and Zdeňka Urešová. 2015. SemEval 2015 task 18: Broad-coverage semantic dependency parsing. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 915–926, Denver, Colorado. Association for Computational Linguistics.

Lilja Øvrelid, Petter Mæhlum, Jeremy Barnes, and Erik Velldal. 2020. A fine-grained sentiment dataset for Norwegian. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5025–5033, Marseille, France. European Language Resources Association.

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 79–86. Association for Computational Linguistics.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. SemEval-2015 task 12: Aspect based sentiment analysis. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 486–495, Denver, Colorado. Association for Computational Linguistics.

Ana-Maria Popescu and Oren Etzioni. 2005. Extracting product features and opinions from reviews. In

Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, pages 339–346, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Soujanya Poria, Erik Cambria, Grégoire Winterstein, and Guang-Bin Huang. 2014. Sentic patterns: Dependency-based rules for concept-level sentiment analysis. *Knowledge-Based Systems*, 69:45–63.

Devendra Sachan, Yuhao Zhang, Peng Qi, and William L. Hamilton. 2021. Do syntax trees help pre-trained transformers extract information? In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2647–2661, Online. Association for Computational Linguistics.

Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Kai Sun, Richong Zhang, Samuel Mensah, Yongyi Mao, and Xudong Liu. 2019. Aspect-level sentiment analysis via convolution over dependency tree. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5679–5688, Hong Kong, China. Association for Computational Linguistics.

Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. 2011. Lexicon-based methods for sentiment analysis. *Computational linguistics*, 37(2):267–307.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. 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 1556–1566, Beijing, China. Association for Computational Linguistics.

Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. 2016. Effective LSTMs for target-dependent sentiment classification. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3298–3307, Osaka, Japan. The COLING 2016 Organizing Committee.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT – building open translation services for the world. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 479–480, Lisboa, Portugal. European Association for Machine Translation.

Cigdem Toprak, Niklas Jakob, and Iryna Gurevych. 2010. Sentence and expression level annotation of opinions in user-generated discourse. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 575–584, Uppsala, Sweden. Association for Computational Linguistics.

Peter Turney. 2002. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 417–424, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

David Vilares, Miguel A. Alonso, and Carlos Gómez-Rodríguez. 2015a. A syntactic approach for opinion mining on Spanish reviews. *Natural Language Engineering*, 21(01):139–163.

David Vilares, Miguel A Alonso, and Carlos Gómez-Rodríguez. 2015b. On the usefulness of lexical and syntactic processing in polarity classification of t witter messages. *Journal of the Association for Information Science and Technology*, 66(9):1799–1816.

David Vilares, Carlos Gómez-Rodríguez, and Miguel A Alonso. 2017. Universal, unsupervised (rule-based), uncovered sentiment analysis. *Knowledge-Based Systems*, 118:45–55.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, 39(2-3):165–210.

Zenan Xu, Daya Guo, Duyu Tang, Qinliang Su, Linjun Shou, Ming Gong, Wanjun Zhong, Xiaojun Quan, Daxin Jiang, and Nan Duan. 2021. Syntax-enhanced pre-trained model. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5412–5422, Online. Association for Computational Linguistics.

Wei Xue and Tao Li. 2018. Aspect based sentiment analysis with gated convolutional networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2514–2523, Melbourne, Australia. Association for Computational Linguistics.

Meishan Zhang, Qiansheng Wang, and Guohong Fu. 2019. End-to-end neural opinion extraction with a transition-based model. *Information Systems*, 80:56–63.

Yuan Zhang and Yue Zhang. 2019. Tree communication models for sentiment analysis. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3518–3527, Florence, Italy. Association for Computational Linguistics.

Z. Zhang, Y. Wu, J. Zhou, S. Duan, H. Zhao, and R. Wang. 2022. Sg-net: Syntax guided transformer for language representation. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (01):1–1.

A Shared Task Leaderboard

User	Team	NoReC_fine	MultiBooked_ca	MultiBooked_eu	OpeNER_en	OpeNER_es	MPQA	Darmstadt_unis	Average
zhixiaobao		0.529 (2)	0.728 (1)	0.739 (1)	0.760 (2)	0.722 (4)	0.447 (1)	0.494 (1)	0.631 (1)
Cong666		0.524 (3)	0.728 (1)	0.739 (1)	0.763 (1)	0.742 (1)	0.416 (2)	0.485 (2)	0.628 (2)
gmorio	Hitachi	0.533 (1)	0.709 (3)	0.715 (3)	0.756 (3)	0.732 (3)	0.402 (3)	0.463 (3)	0.616 (3)
colorful		0.497 (5)	0.678 (6)	0.723 (2)	0.745 (4)	0.735 (2)	0.375 (5)	0.380 (12)	0.590 (4)
whu_stone	sixsixsix	0.483 (9)	0.711 (2)	0.681 (6)	0.727 (6)	0.686 (7)	0.379 (4)	0.373 (13)	0.577 (5)
KE_AI		0.483 (9)	0.711 (2)	0.681 (6)	0.727 (6)	0.686 (7)	0.364 (7)	0.373 (13)	0.575 (6)
Fadi	SeqL	0.488 (7)	0.699 (4)	0.701 (4)	0.730 (5)	0.700 (5)	0.245 (20)	0.394 (11)	0.565 (7)
lys_acoruna	LyS_ACoruña	0.462 (10)	0.653 (8)	0.680 (7)	0.698 (9)	0.692 (6)	0.349 (10)	0.414 (7)	0.564 (8)
QiZhang	ECNU_ICA	0.496 (6)	0.684 (5)	0.686 (5)	0.676 (10)	0.623 (11)	0.351 (8)	0.409 (8)	0.561 (9)
luxinyu	ohhhmygosh	0.487 (8)	0.658 (7)	0.651 (9)	0.710 (7)	0.669 (8)	0.269 (19)	0.416 (6)	0.551 (10)
rafalposwiata	OPI	0.459 (11)	0.650 (9)	0.653 (8)	0.670 (11)	0.663 (9)	0.326 (13)	0.395 (10)	0.545 (11)
evanyfyang		0.213 (22)	0.635 (11)	0.639 (10)	0.703 (8)	0.642 (10)	0.350 (9)	0.449 (4)	0.519 (12)
robvanderg		0.366 (13)	0.648 (10)	0.605 (11)	0.632 (14)	0.614 (13)	0.296 (15)	0.344 (14)	0.501 (13)
psarangi	AMEX AI Labs	0.343 (15)	0.634 (12)	0.559 (12)	0.634 (13)	0.595 (14)	0.283 (17)	0.320 (17)	0.481 (14)
chx.dou	abondoned	0.395 (12)	0.583 (13)	0.506 (13)	0.626 (15)	0.622 (12)	0.309 (14)	0.280 (19)	0.474 (15)
zaizhep	MMAI	0.329 (16)	0.525 (14)	0.478 (17)	0.623 (16)	0.539 (16)	0.367 (6)	0.342 (15)	0.458 (16)
janpf		0.280 (19)	0.517 (15)	0.439 (19)	0.651 (12)	0.504 (17)	0.338 (11)	0.417 (5)	0.449 (17)
etms.kgp	ETMS@IITKGP	0.351 (14)	0.508 (16)	0.438 (20)	0.626 (15)	0.544 (15)	0.327 (12)	0.330 (16)	0.446 (18)
jylong		0.323 (18)	0.474 (19)	0.504 (14)	0.476 (17)	0.375 (21)	0.274 (18)	0.223 (21)	0.379 (19)
ouzh		0.323 (18)	0.474 (19)	0.504 (14)	0.476 (17)	0.375 (21)	0.274 (18)	0.223 (21)	0.378 (20)
SPDB_Innovation_Lab	Innovation Lab	0.325 (17)	0.469 (20)	0.486 (16)	0.471 (18)	0.362 (22)	0.289 (16)	0.202 (22)	0.372 (21)
lucasrafaelc		0.251 (21)	0.505 (17)	0.467 (18)	0.431 (19)	0.399 (19)	0.232 (21)	0.230 (20)	0.359 (22)
foodchup		0.265 (20)	0.493 (18)	0.491 (15)	0.415 (20)	0.480 (18)	0.149 (22)	0.139 (24)	0.347 (23)
jzh1qaz		0.186 (25)	0.431 (21)	0.385 (21)	0.381 (21)	0.393 (20)	0.094 (23)	0.092 (26)	0.280 (24)
hades_d	Mirs	0.504 (4)	0.678 (6)	0.000 (25)	0.000 (25)	0.000 (26)	0.375 (5)	0.400 (9)	0.280 (24)
huyenbui117		0.194 (23)	0.341 (22)	0.374 (22)	0.316 (23)	0.245 (25)	0.009 (26)	0.053 (27)	0.219 (25)
karun842002	SSN_MLRG1	0.191 (24)	0.323 (23)	0.331 (23)	0.306 (24)	0.257 (24)	0.015 (25)	0.104 (25)	0.218 (26)
gerarld	nlp2077	0.000 (26)	0.269 (24)	0.303 (24)	0.354 (22)	0.321 (23)	0.019 (24)	0.180 (23)	0.207 (27)
michael_wzhu91	kobe4ever	0.000 (26)	0.000 (25)	0.000 (25)	0.000 (25)	0.000 (26)	0.000 (27)	0.306 (18)	0.044 (28)
normalkim		0.000 (26)	0.000 (25)	0.000 (25)	0.000 (25)	0.000 (26)	0.000 (27)	0.000 (28)	0.000 (29)
UniParma	UniParma	0.000 (26)	0.000 (25)	0.000 (25)	0.000 (25)	0.000 (26)	0.000 (27)	0.000 (28)	0.000 (29)
whu_venti		0.000 (26)	0.000 (25)	0.000 (25)	0.000 (25)	0.000 (26)	0.000 (27)	0.000 (28)	0.000 (29)

Table 11: Leaderboard of all participants in the monolingual task

User	Team	OpeNER_es	MultiBooked_ca	MultiBooked_eu	Average
Cong666		0.644 (1)	0.643 (1)	0.632 (1)	0.640 (1)
luxinyu	ohhhmygosh	0.620 (3)	0.605 (4)	0.569 (2	0.598 (2)
gmorio	Hitachi	0.628 (2)	0.607 (3)	0.527 (4)	0.587 (3)
whu_stone	sixsixsix	0.604 (5)	0.596 (5)	0.512 (7)	0.571 (4)
QiZhang	ECNU_ICA	0.551 (10)	0.615 (2)	0.530 (3)	0.566 (5)
Fadi	SeqL	0.589 (6)	0.593 (6)	0.516 (6)	0.566 (5)
colorful		0.620 (3)	0.543 (11)	0.527 (4)	0.563 (6)
hades_d	Mirs	0.617 (4)	0.544 (10)	0.522 (5)	0.561 (7)
lys_acoruna	LyS_ACoruña	0.570 (7)	0.554 (8)	0.509 (8)	0.544 (8)
rafalposwiata	OPI	0.564 (8)	0.586 (7)	0.444 (12)	0.531 (9)
KE_AI		0.561 (9)	0.552 (9)	0.463 (11)	0.525 (10)
etms.kgp	ETMS@IITKGP	0.542 (11)	0.506 (12)	0.431 (13)	0.493 (11)
jylong		0.375 (12)	0.474 (13)	0.504 (9)	0.451 (12)
ouzh		0.375 (12)	0.474 (13)	0.504 (9)	0.451 (12)
SPDB_Innovation_Lab	SPDB Innovation Lab	0.362 (13)	0.469 (14)	0.486 (10)	0.439 (13)
gerarld	nlp2077	0.321 (14)	0.269 (15)	0.303 (14)	0.298 (14)
janpf		0.315 (15)	0.259 (16)	0.243 (15)	0.272 (15)
chx.dou	abondoned	0.013 (16)	0.009 (17)	0.004 (16)	0.009 (16)
jzh1qaz		0.000 (17)	0.000 (18)	0.000 (17)	0.000 (17)
zhixiaobao		0.000 (17)	0.000 (18)	0.000 (17)	0.000 (17)
psarangi	AMEX AI Labs	0.000 (17)	0.000 (18)	0.000 (17)	0.000 (17)
normalkim		0.000 (17)	0.000 (18)	0.000 (17)	0.000 (17)
zaizhep	MMAI	0.000 (17)	0.000 (18)	0.000 (17)	0.000 (17)
lucasrafaelc		0.000 (17)	0.000 (18)	0.000 (17)	0.000 (17)
evanyfyang		0.000 (17)	0.000 (18)	0.000 (17)	0.000 (17)
robvanderg		0.000 (17)	0.000 (18)	0.000 (17)	0.000 (17)
michael_wzhu91	kobe4ever	0.000 (17)	0.000 (18)	0.000 (17)	0.000 (17)
UniParma	UniParma	0.000 (17)	0.000 (18)	0.000 (17)	0.000 (17)
huyenbui117		0.000 (17)	0.000 (18)	0.000 (17)	0.000 (17)
karun842002	SSN_MLRG1	0.000 (17)	0.000 (18)	0.000 (17)	0.000 (17)
whu_venti		0.000 (17)	0.000 (18)	0.000 (17)	0.000 (17)
foodchup		0.000 (17)	0.000 (18)	0.000 (17)	0.000 (17)

Table 12: Leaderboard of all participants in the cross-lingual task