Cross-lingual complex word identification with multitask learning

Joachim Bingel
Department of Computer Science
University of Copenhagen, Denmark
bingel@di.ku.dk

Johannes Bjerva
Department of Computer Science
University of Copenhagen, Denmark
bjerva@di.ku.dk

Abstract

We approach the 2018 Shared Task on Complex Word Identification by leveraging a cross-lingual multitask learning approach. Our method is highly language agnostic, as evidenced by the ability of our system to generalize across languages, including languages for which we have no training data. In the shared task, this is the case for French, for which our system achieves the best performance. We further provide a qualitative and quantitative analysis of which words pose problems for our system.

1 Introduction

Complex word identification (CWI) is the task of predicting whether a certain word might be difficult for a reader to understand and is typically used as a first step in (lexical) simplification pipelines (Shardlow, 2014; Paetzold and Specia, 2015, 2016a). This task has received significant attention from the community over the past few years, leading to two shared tasks and several other publications (Shardlow, 2013a,b).

This paper presents our submission to the CWI 2018 shared task (Yimam et al., 2018), at the 13th Workshop on Innovative Use of NLP for Building Educational Applications. This task includes tracks targeting four languages: English, Spanish, German and French. For each of these languages, the task involves prediction of binary labels of whether any of a range of annotators deemed some word or phrase complex, or prediction of the ratio of those who did. The task further differs from previous approaches to CWI in extending the definition of the target units from the word level to multi-word expressions, such that annotations in the training and test set spanned wider stretches of text than single tokens.

Another difference from previous approaches to CWI is that the data is annotated by a mixture of native and non-native speakers, posing an interesting challenge to reconcile the potentially different complexity assessments of these groups.

One challenge in the CWI 2018 shared task is the fact that one of the languages under consideration (French) does not have any training data available. We approach this problem by exploring a combination of multitask learning and cross-lingual learning. In doing so, we aim to answer the following research questions:

RQ 1 How can multitask learning be applied to the task of cross-lingual CWI?

RQ 2 How can complex words be identified in languages which are not seen during training time?

Our contributions also include a thorough qualitative and quantitative error analysis, which shows that long and infrequent words are very likely to be complex, but that non-complex words that display these properties pose a challenge to our system.

2 Related work

2.1 Multitask Learning

Multitask learning (MTL) is the combined learning of several tasks in a single model (Caruana, 1997). This can be beneficial in a number of scenarios. Previous work has shown benefits, e.g., in cases where one has tasks which are closely related to one another (Bjerva, 2017a,b), when one task can help another escape a local minimum (Bingel and Søgaard, 2017), and when one has access to some unsupervised signal which can be beneficial to the task at hand (Rei, 2017). A common approach to MTL is the application of hard parameter sharing, in which some set of parameters in a model is shared between several tasks. We contribute to previous work in MTL by using a hard parameter sharing approach in which

Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications, pages 166–174
New Orleans, Louisiana, June 5, 2018. ©2018 Association for Computational Linguistics

we share intermediate layers between languages, and use one output-layer per language, thus in a sense seeing languages as tasks, similarly to Bjerva (2017a).

2.2 Cross-lingual learning

Cross-lingual learning is the problem of training a model on a given language, and applying it to another (unseen) language. One common approach is to apply cross-lingual word representations, although this has the disadvantage that it tends to place relatively high demands on availability of parallel text. Another frequently used approach in this context is to use machine translation (MT) so as to obtain a monolingual training set (Tiedemann et al., 2014). However, this approach necessarily increases the complexity of a system, as a fully-fledged MT system needs to be incorporated in the pipeline. Furthermore, this approach bypasses the problem of attempting to find methods or feature sets which can be successful across languages. We therefore follow previous work by, e.g. Bjerva and Östling (2017) in that we use hard parameter sharing with language-agnostic input representations. We build upon this by leveraging language-specific resources which are widely available, such as Wikipedia dumps, and WordNet (see Section 5).

2.3 CWI

Automatic complex word identification has a relatively short history as a research task, with first publications including Shardlow (2013a,b)

A noticeable commonality of the highest-scoring systems in the CWI 2016 shared task was the use of ensemble methods, most notably random forest classifiers, which drew on a range of morphologic, semantic and psycholinguistic features, among others (Paetzold and Specia, 2016b; Ronzano et al., 2016).

Yimam et al. (2017) present first work on CWI that considers languages other than English. They release a German and a Spanish dataset and present first CWI results for these languages. Notably, they also describe first cross-lingual experiments, in which they train on some language and test on another, i.e. without employing any of the common strategies for cross-lingual learning that we outline above.

Recently, Bingel et al. (2018) showed promising results in predicting complex words from gaze patterns of Danish children with reading difficulties, which opens up possibilities for personalized complex word identification, but it is less certain how well their method generalizes to other languages or demographics.

Language	Training	Dev	Test	Complex
English	27,299	3,328	4,252	42.03%
Spanish	13,750	1,622	2,233	40.61%
German	6,151	795	959	39.21%
French	–	–	2,251	29.18%

Table 1: Data overview. The share of complex words is computed across all data splits.

3 Data

We use the data made available through the shared task (Yimam et al., 2018). Each training instance consists of a sentence, with a marked complex phrase annotation, including the numbers of native and non-native annotators, and the fraction of these who found the phrase to be complex. An overview of the data is given in Table 1. The number of entries which are considered complex is quite skewed, and differs per language as French has substantially fewer complex phrases than English. This is further illustrated in Figure 1.

In addition to the shared task data, we also use external resources in our feature representations (see Section 5).

4 Model

As outlined in Section 2, earlier work has shown the aptitude of ensemble methods for CWI, especially such ensembles that feature random forests. We further choose to address the problem in a cross-lingual fashion, for which we deem multi-task learning models particularly suitable.

Motivated by these observations, we devise an ensemble that comprises a number of random forests as well as feed-forward neural networks with hard parameter sharing. The random forests each consist of 100 trees that create splits based on Gini impurity (Breiman, 2001). They do not implement any form of explicit cross-lingual transfer other than the reliance on language-agnostic features, such that we simply train them on a single language at a time, or on shuffled concatenations of training data for several languages. We use random forest classifiers for the binary task and random-forest regressors for the probabilistic task.

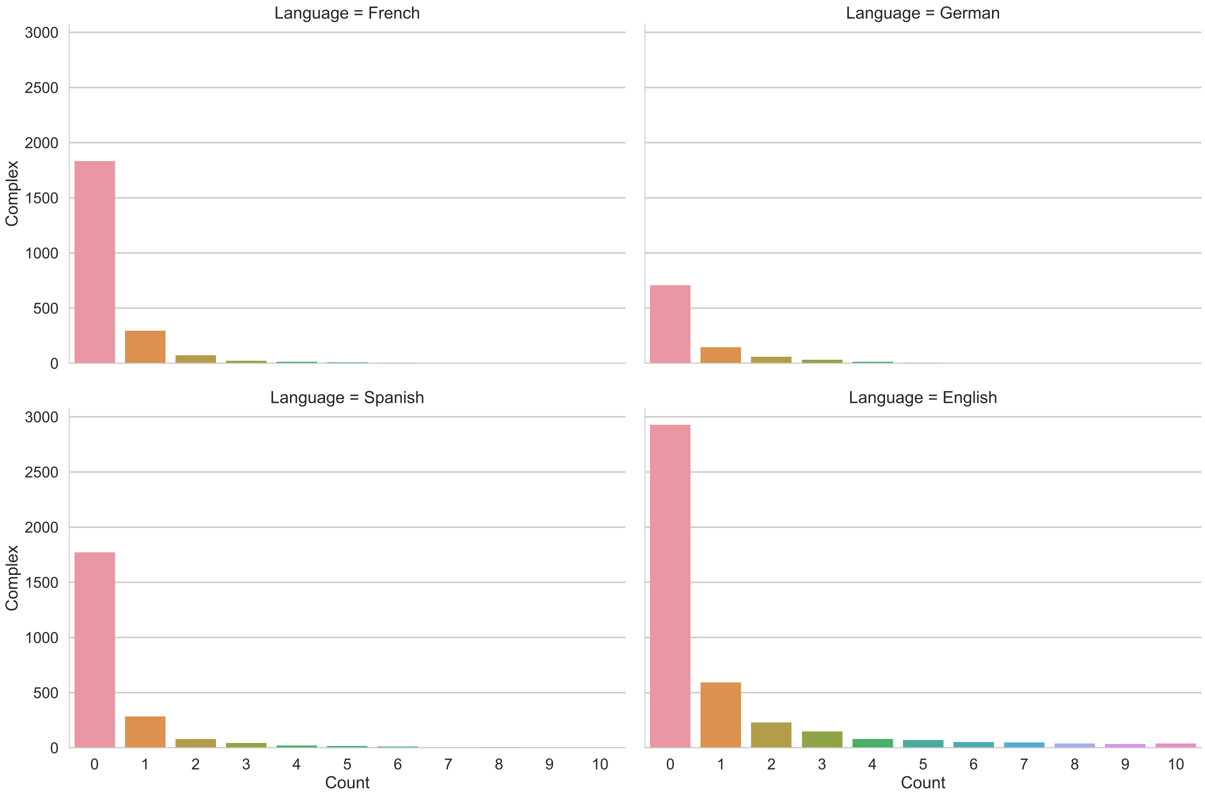


Figure 1: Histogram of numbers sentences (y-axis) which N annotators (x-axis) found to be complex.

Note that our random forests are single-task models, where we cannot define shared or language-specific subparts. Instead, these are always trained on data for the single test language.

The neural MTL models, in contrast, explicitly define parts pertaining to specific languages. Concretely, for each language l, we define a function from a language-specific input layer to a hidden representation h_0 that we share between languages:

$$h_0 = \tanh(x^{(l)} W_{in}^{(l)} + b_{in}^{(l)}) \tag{1}$$

Here and in the following equations, $W_{(\cdot)}$ and $b_{(\cdot)}$ consistently denote weight matrices and bias vectors, respectively. $W_{in}^{(l)}$ and $b_{in}^{(l)}$ are the weights and bias terms specific to input layer l, and the input $x^{(l)}$ is a vector representation of the features introduced in Sec. 5.

We then compute deeper hidden representations, such that the hidden layer at depth d is defined as follows:

$$h_d = \tanh(h_{d-1} W_d + b_d) \tag{2}$$

Finally, each language l defines its own output $y^{(l)}$. This output is defined slightly differently for the regression and classification models.

$$y_{reg}^{(l)} = h_D W_{out}^{(l)} + b_{out}^{(l)} \tag{3}$$

For the former, this is simply a linear transformation of the deepest hidden layer D. The classification model adds a sigmoid activation to this:

$$y_{clf}^{(l)} = \sigma(h_D W_{out}^{(l)} + b_{out}^{(l)}) \tag{4}$$

4.1 MTL training

Since our multitask model defines several outputs, but our data is only labeled with a single annotation layer (i.e. for a single language or “task”), we need a training strategy that does not require true labels for all tasks. The way this is normally approached is to iteratively optimize for tasks in isolation, e.g. by deciding at random which language we sample a batch of data from at every pass of the forward-backward algorithm we use to train the model.

We employ the above strategy and optimize the regression model with a mean absolute error loss function, as well as cross-entropy for the classification model. We monitor these losses on the validation set as an early stopping criterion.

4.2 Ensemble voting

The different neural and random-forest based model that we train as devised above finally make independent predictions for new examples. For the regression models, we use the median prediction across all systems for a given input to make the final ensemble prediction. For the classifiers, however, we have an additional parameter t that we optimize on a held-out development set. This is a threshold indicating the fraction of classifiers that need to cast a positive vote for us to finally accept an example as complex. All neural and random forest classifiers are weighted equally here, each casting a single binary vote.

4.3 Language identification for cross-lingual prediction

As we expect our system to be able to generate predictions for unseen languages (for which we have no explicit output layer), we implement a further component in our neural model that we optimize to predict the language of some input from the set of available languages with explicit output layers. This is an additional output layer of our model, defined as a dense projection from the first hidden layer followed by a sigmoid.

$$l = \sigma(h_0 W_{lid} + b_{lid}) \tag{5}$$

During training, we then supply a ground truth language identifier $\hat{l}$ as a second target variable and perform optimization under a cross-entropy loss that we add to the CWI loss. At test time, for a language without an explicit output layer, we first predict the most similar language we saw during training using Eq. 5 and then use the output layer for that language to generate CWI predictions. An alternative to this could be to generate predictions from all CWI output layers and ensemble these, possibly weighted, with weights inferred in a similar fashion to language identification.

For the random forest models, which do not define language-specific output functions, we simply concatenate training data from all available languages, leveraging the fact that our feature space is language-independent.

5 Features

Our systems build on the same set of features for all input languages, although some of these are computed from language-specific resources. This means that the distributions of values attained for certain features may differ between languages, which is the motivation for language-specific input layers in our model. We further reduce language idiosyncrasies by normalizing all features to the $[0, 1]$ range. The features are listed below.

Log-probability We compute unigram frequencies for candidates as their log-probabilities in language-specific Wikipedia dumps. For multi-word targets, we use the sum of the log-probabilities of the individual items. Log-probabilities are computed using KenLM (Heafield, 2011).

Character perplexity Based on the same Wikipedia dumps as above, we compute character perplexities over the candidate strings using a smoothed 5-gram character-based language model (again using KenLM).

Number of synsets As a measure of a target's semantic ambiguity, we count the number of synsets that include it. For this, we rely on the language-specific WordNet resources for English (Fellbaum, 1998), Spanish (Gonzalez-Agirre et al., 2012) and French (Sagot and Fišer, 2008). For German, access to GermaNet (Hamp and Feldweg, 1997) was harder to obtain, and we instead automatically translate words from German to English and use the English WordNet.[1] In case of a multi-word target, we take the mean number of synsets across the individual words.

Hypernym chain As a measure of semantic specificity, we further consider the length of the hypernym chain of an item, i.e. the number of hypernyms that can recursively be obtained for a word. These are also obtained using WordNet, and again we average over individual words in a target.

Inflectional complexity As a proxy for inflectional complexity (i.e. the number of suffixes appended to a word stem), we measure the difference in length (character count) between the surface form and the stem of a word. For this, we use language-specific instances of the Snowball stemmer (Porter, 2001) as implemented in NLTK (Bird and Loper, 2004).

Surface features As basic surface features, we include the length of an item (in characters) and whether it is all-lowercase.

[1] For the translations, we used a bilingual dictionary (https://www.dict.cc/).

Language	MAE	Rank	Δ (system)	F_1	Rank	Δ (system)
French	0.066	1	0.012 (TMU)	0.7595	1	0.013 (TMU)
German	0.075	2	-0.013 (TMU)	0.6621	5	-0.083 (TMU)
Spanish	0.079	3	-0.007 (TMU)	0.7458	5	-0.024 (TMU)

Table 2: Official performance figures of our method for all non-English tracks. The Δ (system) column indicates the difference in performance between our system and the best system in each track except for ours. In accordance with the shared task report, classification performance is measured by macro F_1 between the complex and non-complex class in the official results.

Bag-of-POS For each tag defined in the Universal Part-of-Speech project (Petrov et al., 2011), we count the number of words in a candidate that belong to the respective class. We obtain POS tags from spacy.[2]

Target-sentence similarity Motivated by the conjecture that words or phrases are easier to understand if they display higher semantic similarity with their context, we compute the cosine distances between averaged word embeddings for the target word or phrase and the rest of the containing sentence. To mitigate out-of-vocabulary problems, we use pretrained subword embeddings that we retrieve from the BPEemb project (Heinzerling and Strube, 2017).

The data provided for the shared task further includes information on how many of the annotators are native and non-native speakers. While this information is potentially helpful (assuming that non-native speakers would have a stronger bias for annotating as complex), we do not make use of it, considering that access to such information cannot be assumed in a real-world scenario.

6 Results

We present an overview of the results that our method (as well as our best contender) achieved in Table 2 and discuss results for the individual languages below.[3]

6.1 French

Due to the lack of training data for this track, it poses a challenging test for the ability of our models to generalize across languages. While the exact performance figures are at least partly subject to idiosyncrasies in the text samples and annotators, the results obtained here are a good benchmark of of what we can achieve for languages for which we do not even have validation data to monitor development loss for early stopping.

As Table 2 shows, we achieve the best results of all participating teams for French, both for the classification and for the regression track. We view this as evidence that our cross-lingual MTL approach is an effective means to share knowledge between different data sources and even different languages.

6.2 German/Spanish

Our results for Spanish and German show that, while we did not achieve the best results compared to other participants, our method still performs competitively. Especially for the regression track, while not ranking first, the absolute performance figures place us very close to the winning systems. We see this as a validation of our approach, in particular under the consideration that a gradual assessment of complexity is perhaps more meaningful than a binary one, especially when the definition of the latter makes no distinction between one or all out of 20 annotators judging an item as difficult.

6.3 Analysis

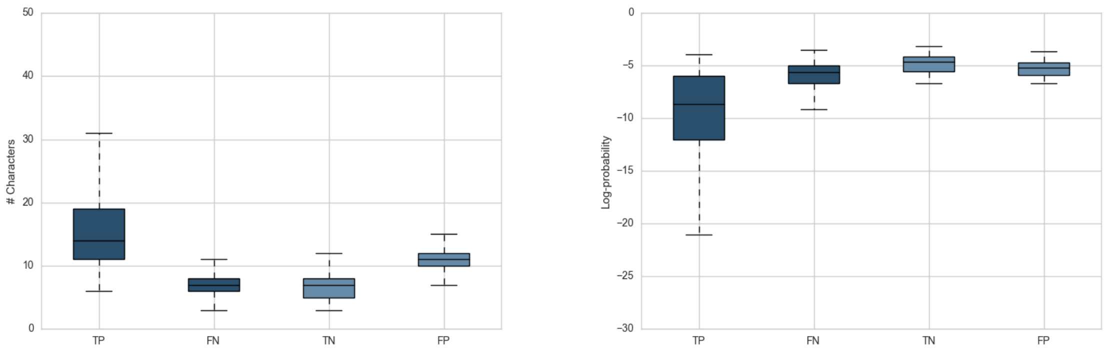
Qualitative error analysis Table 3 exemplifies some of the correct and incorrect predictions that our system makes for the French test data. We observe that the system picks up on the relatively long targets listed as true positives. Note also that the false positives are relatively long words, which suggests that the system is deceived by this. The targets listed as false negatives are shorter, but they are examples of a (potentially unknown) named entity and a relatively technical term, which might pose difficulties to some readers. The words listed as true negatives are correctly predicted by our

[2] `https://spacy.io/`

[3] We did not submit solutions for the English track.

True positives
Il **marque néanmoins sa désapprobation** en voyant des Juifs prier devant le mur des Lamentations; Einstein commente qu'il s'agit de personnes collées au passé et faisant abstraction du présent.
Rimbaud a donné ses lettres de noblesse à un type de poème en prose distinct d'expériences plus **prosaïques** du type du "Spleen de Paris" de Baudelaire.
False negatives
Le pays des vallées d'Andorre entre la France et l'Espagne, sur le versant sud des **Pyrénées**, est constitué par deux vallées principales: celle du Valira del Orient et celle du Valira del Nord dont les eaux réunies forment le Valira.
Autres cultures permanentes, la lavande et le lavandin occupent plusieurs milliers d'**hectares** et fournissent plusieurs milliers d'emplois directs.
True negatives
Beaucoup d'îles des Caraïbes (les Antilles) – par exemple, les Grandes Antilles et les Petites Antilles – sont **situées** au-dessus de la plaque caraïbe, une plaque tectonique avec une topographie diffuse.
Avec un fort penchant à l'hermétisme qu'il partage avec d'autres de ses quasi contemporains (Gérard de Nerval, Stéphane Mallarmé, sinon Paul Verlaine parfois), Rimbaud a le **génie** des visions saisissantes qui semblent défier tout ordre de description du réel.
False positives
La **construction** de l'Atomium fut une prouesse technique.
La **proportion** des musulmans, tous sunnites, est inférieure à 1%.

Table 3: Example wins and losses of our model for French. Target words or phrases are marked in bold.



(a) Length in characters per error type

(b) Log-probability per error type

Figure 2: Statistics of character length and language model log-probability for the French test set. The darker-shaded boxes are complex words that we predicted correctly (TP) and incorrectly (FN), respectively. The lighter-shaded boxes are non-complex words, predicted correctly (TN) and incorrectly (FP).

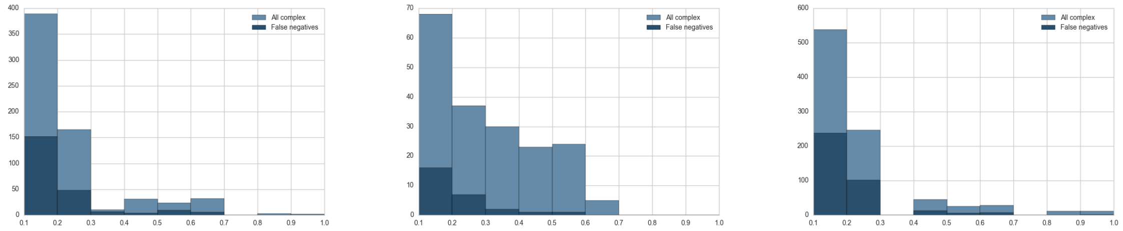


Figure 3: Distributions of false negative predictions per complexity degree as measured by the fraction of annotators that labeled items as complex in the French, German and Spanish test sets (left to right).

system as non-complex, possibly because of their shortness.

Quantitative error analysis Investigating the observations from the previous section in a more quantitative fashion, Figure 2 presents distributions of two basic features across complex vs. non-complex words, and correctly vs. wrongly predicted test set items for French. For item length, we observe a clear pattern that complex words tend to be significantly longer than non-complex ones. Further, the longer they are, the easier it is for our model to detect them as complex. Non-complex words that are relatively long, however, lead to incorrect predictions from our model.

A very similar pattern can be observed for the log-probability of complex and non-complex items. The former are assigned a much lower probability by our language model, and particularly unlikely words are very easy to detect as complex. In turn, non-complex words with relatively low probability pose a challenge for our model.

False negatives per complexity degree We further analyze the influence of the degree of complexity on our model's ability to detect complex words. As stated in the Introduction, an item is labeled as complex in the classification setting if any of the annotators deemed it to be complex. Effectively, no distinction is made in the classification task between a "slightly complex" item that was marked as such by just one out of ten annotators, and one that was unanimously considered complex.

A natural assumption is that our models would more easily pick up on highly complex words and predict false negatives mostly for items with low complex annotation ratios. To verify this assumption, we plot the total number of complex words in the three non-English test sets against the false negative predictions of our model, grouped by the ratio of annotators who marked an item as complex (Figure 3). The French and Spanish test sets are somewhat inconclusive for our question as they generally contain very few items with a complexity ratio higher than 0.2. The German test set, however, is more balanced, and in fact we observe that items with a complexity ratio above 0.2 are very reliably detected by our model, confirming our hypothesis.

7 Discussion

We approached **RQ 1** by using one output layer per language, and sharing intermediate parameters. This approach was successful, at least in part due to our language-agnostic input representations, which allowed the model to learn similar internal representations for each language. Separate output-layers per language, in turn, allow for the model to make language-specific accommodations. We approached **RQ 2** by using language-agnostic feature representations, and language-specific output layers which were chosen during test time for unseen languages. This approach allowed our model to perform well on the unseen language French, and in fact outperformed our results on other languages. This is, however, not strictly a fair comparison as it is possible that the French test set was somehow easier than the others.

8 Conclusion

We tackled the 2018 Shared Task on CWI with a cross-lingual approach via multitask learning. Our system is highly language-agnostic, as evidenced by our high performance on French, which was not seen during training time. Our analysis confirms that word length and frequency are good, crosslinguistic predictors of complexity. However, the concrete relationships between these features and complexity may differ between languages, which is captured by our multitask learning approach. Our approach is especially promising for the application of CWI to unseen languages, as we do not assume access to any target language training data. Furthermore, this could even substantially facilitate the creation of new CWI datasets, using a bootstrapping or active learning approach.

Acknowledgments

James Thorne wrote much of the Pytorch code that we used in this work. We are grateful to the anonymous reviewer for helpful suggestions. The first author acknowledges the support by Trygfonden.

References

Joachim Bingel, Maria Barrett, and Sigrid Klerke. 2018. Predicting misreadings from gaze in children with reading difficulties. In *Proceedings of the 13th Workshop on Innovative Use of NLP for Building Educational Applications*, New Orleans, United States. Association for Computational Linguistics.

Joachim Bingel and Anders Søgaard. 2017. Identifying beneficial task relations for multi-task learning in deep neural networks. *15th Conference of the European Chapter of the Association for Computational Linguistics*.

Steven Bird and Edward Loper. 2004. Nltk: the natural language toolkit. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, page 31. Association for Computational Linguistics.

Johannes Bjerva. 2017a. *One Model to Rule them all: Multitask and Multilingual Modelling for Lexical Analysis*. Ph.D. thesis, University of Groningen.

Johannes Bjerva. 2017b. Will my auxiliary tagging task help? Estimating Auxiliary Tasks Effectivity in Multi-Task Learning. In *Proceedings of the 21st Nordic Conference on Computational Linguistics, NoDaLiDa, 22-24 May 2017, Gothenburg, Sweden*, 131, pages 216–220. Linköping University Electronic Press, Linköpings universitet.

Johannes Bjerva and Robert Östling. 2017. Cross-lingual Learning of Semantic Textual Similarity with Multilingual Word Representations. In *Proceedings of the 21st Nordic Conference on Computational Linguistics, NoDaLiDa, 22-24 May 2017, Gothenburg, Sweden*, 131, pages 211–215. Linköping University Electronic Press, Linköpings universitet.

Leo Breiman. 2001. Random forests. *Machine learning*, 45(1):5–32.

Rich Caruana. 1997. Multitask learning. *Machine Learning*, 28 (1):41–75.

Christiane Fellbaum. 1998. A semantic network of english verbs. *WordNet: An electronic lexical database*, 3:153–178.

Aitor Gonzalez-Agirre, Egoitz Laparra, and German Rigau. 2012. Multilingual central repository version 3.0. In *LREC*, pages 2525–2529.

Birgit Hamp and Helmut Feldweg. 1997. Germanet-a lexical-semantic net for German. *Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*.

Kenneth Heafield. 2011. Kenlm: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197. Association for Computational Linguistics.

Benjamin Heinzerling and Michael Strube. 2017. Bpemb: Tokenization-free pre-trained subword embeddings in 275 languages. *arXiv preprint arXiv:1710.02187*.

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

Gustavo Paetzold and Lucia Specia. 2016a. Semeval 2016 task 11: Complex word identification. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 560–569.

Gustavo Paetzold and Lucia Specia. 2016b. Sv000gg at semeval-2016 task 11: Heavy gauge complex word identification with system voting. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 969–974.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2011. A universal part-of-speech tagset. *arXiv preprint arXiv:1104.2086*.

Martin F Porter. 2001. Snowball: A language for stemming algorithms.

Marek Rei. 2017. Semi-supervised multitask learning for sequence labeling. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2121–2130. Association for Computational Linguistics.

Francesco Ronzano, Luis Espinosa Anke, Horacio Saggion, et al. 2016. Taln at semeval-2016 task 11: Modelling complex words by contextual, lexical and semantic features. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1011–1016.

Benoît Sagot and Darja Fišer. 2008. Building a free french wordnet from multilingual resources. In *OntoLex*.

Matthew Shardlow. 2013a. A comparison of techniques to automatically identify complex words. In *51st Annual Meeting of the Association for Computational Linguistics Proceedings of the Student Research Workshop*, pages 103–109.

Matthew Shardlow. 2013b. The cw corpus: A new resource for evaluating the identification of complex words. In *Proceedings of the Second Workshop on Predicting and Improving Text Readability for Target Reader Populations*, pages 69–77.

Matthew Shardlow. 2014. Out in the open: Finding and categorising errors in the lexical simplification pipeline. In *LREC*, pages 1583–1590.

Jörg Tiedemann, Željko Agić, and Joakim Nivre. 2014. Treebank translation for cross-lingual parser induction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 130–140. Association for Computational Linguistics.

Seid Muhie Yimam, Chris Biemann, Shervin Malmasi, Gustavo Paetzold, Lucia Specia, Sanja Štajner, Anaïs Tack, and Marcos Zampieri. 2018. A Report on the Complex Word Identification Shared Task 2018. In *Proceedings of the 13th Workshop on Innovative Use of NLP for Building Educational Applications*, New Orleans, United States. Association for Computational Linguistics.

Seid Muhie Yimam, Sanja Štajner, Martin Riedl, and Chris Biemann. 2017. Multilingual and cross-lingual complex word identification. In *Proceedings of RANLP*, pages 813–822.