JUSTMasters at SemEval-2020 Task 3: Multilingual Deep Learning Model to Predict the Effect of Context in Word Similarity

Nour Al-Khdour
Dept. of Computer Science
Jordan University of Science
and Technology-Irbid, Jordan
*n*aalkhdour17@cit.just.edu.jo

Mutaz Bni Younes
Dept. of Computer Science
Jordan University of Science
and Technology-Irbid, Jordan
*m*mbniyounes18@cit.just.edu.jo

Malak Abdullah
Dept. of Computer Science
Jordan University of Science
and Technology-Irbid, Jordan
*m*abdullah@just.edu.jo

Mohammad AL-Smadi
Dept. of Computer Science
Jordan University of Science
and Technology-Irbid, Jordan
*m*aalsmadi9@just.edu.jo

Abstract

There is a growing research interest in studying word similarity. Without a doubt, two similar words in a context may be considered different in another context. Therefore, this paper investigates the effect of the context in word similarity. The SemEval-2020 workshop has provided a shared task (Task 3: Predicting the (Graded) Effect of Context in Word Similarity). In this task, the organizers provided unlabeled datasets for four languages, English, Croatian, Finnish, and Slovenian. Our team, JUSTMasters, has participated in this competition in the two subtasks: A and B. Our approach has used a weighted average ensembling method for different pre-trained embeddings techniques for each of the four languages. Our proposed model outperformed the baseline models in both subtasks and achieved the best result for subtask 2 in English and Finnish, with a score of 0.725 and 0.68, respectively. We have been ranked the sixth in subtask 1, with English, Croatian, Finnish, and Slovenian with scores as follows: 0.738, 0.44, 0.546, 0.512.

1 Introduction

Identifying and measuring words similarity is considered the base step for spotting the similarity between sentences or documents. Word similarity is classified into three types: the first type is lexical similarity (Gomaa et al., 2013), which depends on the symmetry of the character sequence; the second one is semantic similarity (Miller and Charles, 1991), which depends on the literal meaning of the words; and the third type is pragmatics (Cherpas, 1992), which focuses on the meanings and the effects of words according to the context.

Task 3 in SemEval 2020 (Armendariz et al., 2020a) is about finding the context effect on word meanings by giving two similar words in a different context. Task 3 has two unsupervised subtasks, the first subtask asked to predict the degree and direction of change in the human annotator's scores of similarity between two words when presented in two different contexts. The second subtask asked to predict the human score of similarity for a pair of words within the same context. In addition, the task provided data for four different languages: English, Croatian, Finnish, and Slovenian.

In this paper, we have experimented with different embeddings techniques in each of the four languages to determine the best combination for each subtask. Also, we propose a weighted ensembling for state-of-the-art embedding techniques to achieve a significant sense of representation of the words. This has been accomplished for the four languages to gain the highest result in both subtasks except the English language. Besides, our proposed model has outperformed the other teams in subtask 2 for English and Finnish, with the highest score of 0.725 and 0.68 respectively.

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 292–300
Barcelona, Spain (Online), December 12, 2020.

This paper is organized as follows: related work will be shown in section 2. Section 3 provides a brief description of Task3 and the dataset. Our proposed model, JUSTMasters, is characterized in Section 4. Experiments and results are introduced in section 5. Finally, the conclusion is provided in section 6.

2 Related Work

Different similarity measurements are used to determine the similarity between texts in various ways. There are four known categories of text similarity measurements: string-based, knowledge-based, corpus-based, and hybrid-based (Prasetya et al., 2018). The researchers agreed that string-based similarity algorithms are the best tool for measuring the lexical similarity, and Knowledge-based similarity algorithms are the best for the semantic similarity.

In order to estimate the score of similarity between sentences, researchers in (Mueller and Thyagarajan, 2016) applied Manhattan Long Short-Term Memory (MaLSTM) with Support Vector Machine classifier by using 300-dimensions Word2vec as word embedding. Moreover, to capture the semantic similarity between fixed-length texts, a group of researchers (De Boom et al., 2015) proposed a hybrid technique that consists of TF-IDF (Joachims, 1996) and word embedding. TF-IDF technique performs well with logical similarity, while the word embedding performs well with semantic similarity, so they combined the previous technique to classify the texts using Euclidean distance. Other researchers (Recski et al., 2016) used a novel combination of word embeddings, WordNet, and the concept dictionary 4lang to measure the semantic similarity of word pairs.

The authors of (He and Lin, 2016) presented a novel approach consisted of Bi-LSTM and pairwise word interaction modeling techniques to compare each word with the overall sentence. On the other hand, for traditional word embeddings, other researchers (Jatnika et al., 2019) showed that the best values for the Word2vec features are 9 as window size and 300 as vector dimensions configuration.

For determining the sense of the words based on the contexts, the authors of (Levine et al., 2019) proposed a model called SenseBERT, which predicts the sense in the semantic level of the missing words based on self-predictions of WordNet supersense (Fellbaum, 2012). As well for Word in Context task, SenseBERT was attaining a state of the art result. For semantic relationships in semantic frame induction, a group of researchers (Arefyev et al., 2019) proposed a combined approach that employed two different types of vector representations: dense representations from BERT hidden layers of a masked language model, and sparse representations based on substitutes for the target word in the context.

3 Task Description

Taks 3 in SemEval 2020 consisted of two subtasks:

- **Subtask 1:** It aims to determine the difference between the similarity of two words in different contexts using Equation 1. For example, by giving the words "People" and "Population" in two different contexts, it is noticed that in the second context, both words are highly similar, while the similarity for the same words is decreased in the first context as presented in Table1.

$$change = \textit{sim\_context2} - \textit{sim\_context1} \tag{1}$$

- **Subtask 2:** It aims to find the degree of similarity between two words in different contexts. We need to find the similarity between these two words on a scale of 0-10. In the same example presented in Table1, giving the words "People" and "Population", the similarity score in context 1 is 7.46, and 9.7 for context 2.

Data Description: Task 3 dataset is called CoSimlex (Armendariz et al., 2020b). It was released in four languages: English, Croatian, Finnish, and Slovenian. The English dataset contains pairs of words derived from SimLex-999 (Hill et al., 2015), while for Croatian, Finnish, and Slovenian the pairs are translated from English to each language. Moreover, it contains two different contexts extracted from Wikipedia, each context has a pair of words. The size of each language: 340 English pairs, 112 Croatian pairs, 111 Slovenian pairs, and 24 Finnish pairs.

Context 1	The tornado was also the second costliest tornado in history at the time, but in the years since has been surpassed by several others if **population** changes over time are not considered. When costs are normalized for wealth and inflation, it ranks third today. The deadliest tornado in world history was the Daultipur-Salturia Tornado in Bangladesh on April 26, 1989, which killed approximately 1300 **people**.
Context 2	Agriculture contributes over 45% to the net state domestic product. It is the main source of income and employment in Himachal. Over 93% of the **population** in Himachal depend directly uponagriculture which provides direct employment to 71% of its **people**.

Table 1: Example word pair in two different contexts

Evaluation Metric: The evaluation metric used for the first subtask is an uncentered Pearson correlation against gold scores by human annotators. While for the second task, the evaluation metric is the harmonic mean of the Pearson and Spearman correlations against the gold scores by human annotators.

Baseline: The baseline model provided by the organizers was a BASE Bidirectional Encoder Representations from Transformers (BERT) using the pre-trained 'wiki_multilingual_uncased' model from Google, and using bert_embedding package (Imgarylai, 2019), which is an open-source project to generate token embeddings with multilingual BERT. Table 2 illustrates the baseline results for each task.

Task	**English**	**Finnish**	**Croatian**	**Slovenian**
1	0.7125	0.6712	0.5872	0.6034
2	0.57312	0.2894	0.4024	0.5159

Table 2: The results for baseline model

4 JUSTMasters Approach

Our approach goes in the following work-flow, as shown in **Figure. 1**: the input consists of the required context, if the context's language is English, feature extraction is applied as a further step, otherwise, the next step is word embedding. The embeddings feed our proposed model, JUSTMaster Model. Finally, the evaluation step is done using the Pearson correlation for subtask 1, and harmonic mean of the Pearson and Spearman correlations for subtask 2. Further details is provided in the following subsections.

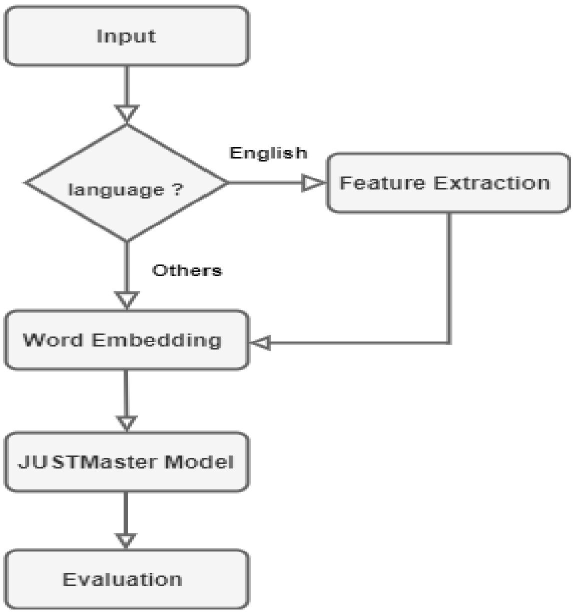

Figure 1: General Work-flow

4.1 Cleaning and Preprocessing Data

We have noticed that cleaning and preprocessing the data lead to a low score result. We believe that this is due to the used contextual embedding. Except removing <strong> and </strong> which surround the pair of words in each context. The experimented cleaning and preprocessing methods consisted of: removing stop words, punctuation, digits, hashtags, and special characters. Also, set all words to lowercase.

4.2 Feature Extraction

We have extracted features related to Natural Language Processing (NLP) field. We have experimented with a list of features, but only eight features shown exceptional results, as follows: **Term Frequency–Inverse Document Frequency** (TF-IDF) feature (Joachims, 1996) that is a numerical statistic number to represent the importance of each word from the pair in both contexts. **Synonyms** feature (Turney, 2008) to find if both words in pair are synonyms or not. **Antonyms** feature (Turney, 2008) to find if both words in pair are antonyms or not. **Edit Distance** feature (Ristad and Yianilos, 1998) is also called Levenshtein Distance, it measures the number of edit moves needed to make two words the same. **Euclidean Distance** (Cha, 2007) is the most common simple measure that calculates the length of the path connecting between two points. **minkowski distance** (Boriah et al., 2008) is a generalization of Euclidean Distance to measure the similarity between two words. **Morphological** feature (Bender, 2013) that is used to study the surface form and grammatical structure of words. **Named Entity Recognition** (NER) feature (Lample et al., 2016) to recognize the type of entities such as locations, products, organizations, person names, etc.

These features are extracted using different Python libraries: NLTK (Bird et al., 2009) for edit distance, antonyms, and synonyms, sklearn (Pedregosa et al., 2011) for TF-IDF, and spaCy (Honnibal and Montani, 2017) for NER and morphological features.

4.3 Embedding techniques

Embedding techniques are mainly converting words to a vector of numbers. Our experiments have been focused on task 2, because task 1 results can be extracted from task 2 results. This sub section provides a list of the methods that have been tested with all languages. We have used well known embedding techniques, we have used weighted combination of embedding. We will illustrate variety of embedding models that we have used for each language.

- **Word Embeddings**: Classic word embeddings are static at word-level, which means that each word has one pre-computed embedding. There are different pre-trained embeddings types under this package (GloVe, fastText embeddings trained over Wikipedia, and fastText embeddings over Web crawls). Through our experiments, we have implemented word embeddings that are supported from Flair library (Akbik et al., 2018). Word Embeddings support most of the datasets language, for example 'en' for English, 'fi' for Finnish, 'sl' for Slovenian, and 'hr' for Croatian. In our experiment, we used 2 types of the supported WordEmbedding, fastText embeddings trained over Wikipedia models initialized as 'en' for example, and FastText embeddings over web crawls initialized as 'en-crawl' for example.

- **fastText**: FastText Embeddings provide vectors for Out Of Vocabulary (oov) (Pinter et al., 2017) words by using the sub-word information. These models (Grave et al., 2018) were trained on Common Crawl and Wikipedia. fastText uses CBOW with position-weights, and has 300 dimension, with character n-grams of length 5, a window of size 5 and 10 negatives. Through our experiments, we have implemented fastText Embeddings (Bojanowski et al., 2017) that is supported from Flair library. FastText is pre-trained word vectors for 157 languages, and among them (English, Finnish, Slovenian, and Croatian).

- **BERT**: BERT is state-of-the-art in the NLP field due to its high performance and highly pragmatic approach. BERT overcomes the shortcoming of other traditional embeddings that provide the same vector for a word regardless of the context, which reduces the importance of the context influence in determining the meaning or the sense of a word. To resolve this dilemma, BERT uses a bidirectional transformer (Vaswani et al., 2017) to provide word embeddings, and it does take into consideration the context and the surrounding words of the word. This gives us two different embedding vectors for the same word in different context. In our experiment we implemented BertEmbeddings package (Imgarylai, 2019) to obtain the token embedding from BERT's pre-trained model.

- **ELMo**: ELMo is a contextual word embedding that was presented in 2018 (Peters et al., 2018). It is a deep contextualized word representation that models both complex characteristics of word use, and how these uses vary across linguistic contexts.
- **RoBERTa**: Facebook proposed modifications to the BERT pretraining procedure that improved end-task performance. They aggregated these improvements and evaluated their combined impact. RoBERTa (Liu et al., 2019) stands for Robustly optimized BERT approach.
- **Flair**: Flair embedding (Akbik et al., 2018) produced the word embedding based on contextual string embeddings, which can capture the latent syntactic-semantic information. It provides different options to choose the embedding, such as 'forward' or 'backward' or Stacked Embeddings for both of them.

4.4 JUSTMaster Architecture

For each type of the four languages, we have built a model to adapt the required language features. **Figure. 2** illustrates the architecture of the English dataset's model for context 1, as well it is applied to context 2. This model extracts features from the context and the embeddings using two different embedding technique (BERT embedding (Devlin et al., 2019) and RoBERTa (Liu et al., 2019)) for the pair of words. Then, we have concatenated the extracted features from the previous step with BERT embeddings. Subsequently, the model adopts a weighted combination between the embeddings for the English language, as 0.7 weight for BERT embedding, and 0.65 weight for RoBERTa. We experimented with weights randomly, 0.7, and 0.65 was the best to obtain a similarity score nearly the human scores of similarity. After that, we have concatenated the weighted vector and calculate the similarity using a cosine similarity measure.

Regarding Finnish dataset, we have built a slightly different model. **Figure. 3** illustrates the architecture of the JUSTMasters model for the Finnish dataset model for context 1, as well it is applied to context 2. We have concatenated the the embeddings of BERT Embedding ('bert-base-finnish-cased-v1') (Devlin et al., 2019) multiplied by 0.7 and Classic WordEmbedding (Akbik et al., 2018) multiplied by 0.65 as a weighted concatenation. Then, the similarity between the pair of words in each context is measured by Cosine similarity measure.

Concerning Slovenian and Croatian, we have implemented the same weighted embedding approach that is used for Finnish but with slight differences in the embedding techniques. For the Croatian language, we have used WordEmbeddings ('hr-crawl') (Akbik et al., 2018) weighted by 0.7 and fasttext (Bojanowski et al., 2017) by 0.65. And, for the Slovenian language, we have adopted our JUSTMasters model using WordEmbeddings('sl') (Akbik et al., 2018) weighted by 0.7 and fasttext (Bojanowski et al., 2017)by 0.65.

5 Results

In this section, we will talk about the results of each language individually. The scores are based on subtask 2.

5.1 English Language

Table 3 shows our experiments on this language for subtask 2. We can see that when we concatenate BERT LARGE (book corpus wiki en uncased) embeddings with Roberta-base embeddings we obtain high scores, which is more than 0.71. However, when we add the extracted features to the equation, the score becomes 0.725, which ranks at the top of the leaderboard. We used the same approach for subtask1 that also has achieved 0.738 scores, and we have ranked sixth on the leaderboard.

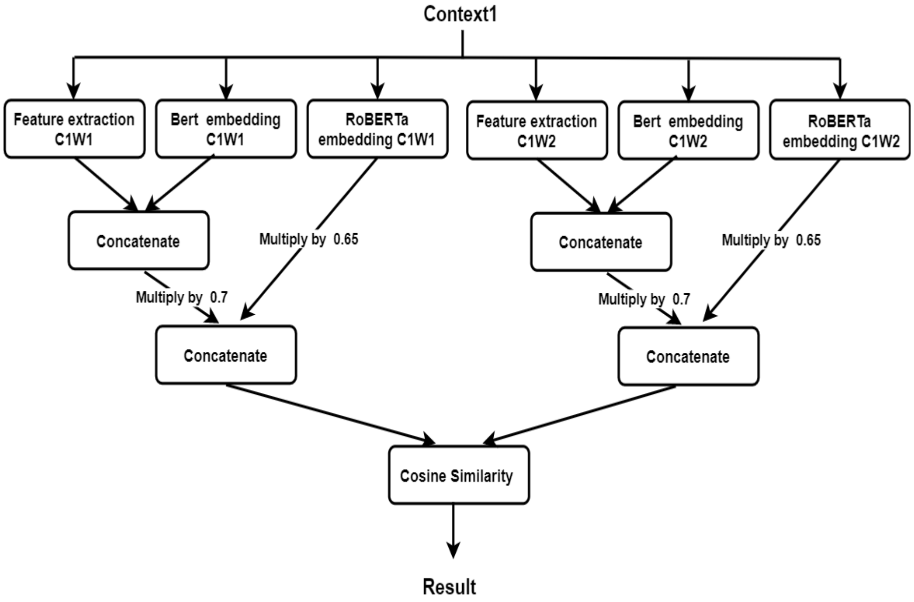

Figure 2: The Architecture of JUSTMaster Model for each Context in English language

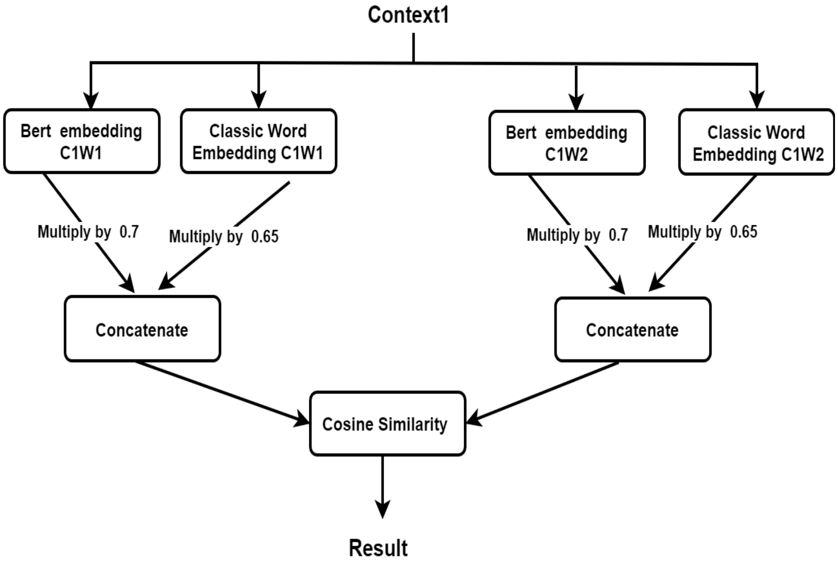

Figure 3: The Architecture of JUSTMaster Model for each Context in Finnish language

Embedding	Score
JUSTMasters model(BERT* + Roberta ('base') + Features)	0.725
JUSTMasters model(BERT* + Roberta ('base'))	0.71
RoBERTa ('base')	0.56
ELMo ('original')	0.42
BERT* LARGE(book corpus wiki en uncased)	0.69
Baseline (BERT-base-wiki-multilingual-uncased)	0.57

Table 3: Embedding models with the English dataset

5.2 Slovenian Language

For the Slovenian language, Table 4 shows our experiments on this language for subtask 2. As the table shows, the traditional word embeddings technique is considered good enough to overcome the baseline, even the results are close, the classic word Embeddings ('sl') achieved 0.52 score which is similar to BERT-base-multilingual-uncased. Also, we have implemented the JUSTMasters model using WordEmbeddings('sl') weighted by 0.7 and fasttext by 0.65, the result for subtask 2 is 0.44, and for subtask 1 is 0.512.

Embedding	**Score**
WordEmbeddings('sl')	0.52
stacked(Flair ('sl-forward') + WordEmbeddings('sl')	0.46
JUSTMasters model (WordEmbeddings('sl') + fasttext('sl'))	0.44
BERT-base-multilingual-uncased	0.52
Baseline (BERT-base-wiki-multilingual-uncased)	0.51

Table 4: Embedding models with the Slovenian dataset

5.3 Croatian Language

For the Croatian language, Table 5 shows our experiments on this language for subtask 2. Also, in this language, the traditional Word embedding technique is good enough to overcome the baseline, for subtask 2 the JUSTMasters model achieved a 0.443 score through using WordEmbeddings('hr') weighted by 0.70 and fasttext weighted by 0.65. Using the same method for subtask 1, we obtained a 0.440 score.

Embedding	**Score**
WordEmbeddings('hr-crawl')	0.532
Stacked(Flair 'hr-forward' + Flair 'hr-backward')	0.301
JUSTMasters model (WordEmbeddings('hr') + fasttext ('hr'))	0.443
BERT-base-multilingual-uncased	0.338
Baseline (BERT-base-wiki-multilingual-uncased)	0.402

Table 5: Embedding models with the Croatian dataset

5.4 Finnish Language

We obtained the best results among all teams on the contest with the Finnish language, too. Table 6 shows our experiments on this language for subtask 2. There is a pre-trained BERT model for this language that gave us a better score than using the multilingual BERT model. However, to get a higher score we used the JUSTMasters model. We gave weight for the word embedding vector and weight for the BERT vector. This technique gained 0.68 scores, #1 on the leaderboard for subtask 2, while 0.546 scores for subtask 1.

embedding model	**Score**
WordEmbeddings('fi')	0.52
stacked(Flair ('fi-forward') + Word embedding ('fi'))	0.55
BERT-base-finnish-uncased-v1	0.58
stacked(WordEmbeddings('fi') + Flair ('fi-forward') + BERT-base-finnish-uncased-v1)	0.59
JUSTMasters model (BERT-base-finnish-uncased-v1 + WordEmbeddings('fi'))	0.68
Baseline (BERT-base-wiki-multilingual-uncased)	0.29

Table 6: Embedding models with the Finnish dataset

6 Conclusion

In this paper, we describe our proposed models for SemEval Task 3. The task is designed for multilingual datasets that include English, Croatian, Finnish, and Slovenian datasets. We also presented the experiments of our model with each language. JUSTMaster model achieved excellent results in the competition on the leaderboard for English and Finnish languages. The model for the English language consists of a concatenation of BERT embedding, Roberta embedding, and the extracted features, which achieved a 0.725 score for subtask 2, and 0.738 for subtask 1. While for the Finnish language, we adopted the JUSTMasters model to concatenate the BERT and traditional word embedding vectors that achieved 0.68 scores for subtask 2 and a 0.546 score for subtask 1. On the other hand, for Slovenian and Croatian languages, we implemented the JUSTMasters model with the traditional Word Embeddings, which obtained 0.44 and 0.443 respectively for subtask 2 and 0.512 and 0.440 respectively for subtask 1. It is worth mentioning that the extracted features such as TFIDF, POS, and other features from both words in each context increased the overall accuracy by 0.15%.

References

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *COLING 2018, 27th International Conference on Computational Linguistics*, pages 1638–1649.

Nikolay Arefyev, Boris Sheludko, Adis Davletov, Dmitry Kharchev, Alex Nevidomsky, and Alexander Panchenko. 2019. Neural granny at semeval-2019 task 2: A combined approach for better modeling of semantic relationships in semantic frame induction. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 31–38.

Carlos S. Armendariz, Matthew Purver, Senja Pollak, Nikola Ljubešić, Matej Ulčar, Marko Robnik-Šikonja, Ivan Vulić, and Mohammad Taher Pilehvar. 2020a. SemEval-2020 task 3: Graded word similarity in context (GWSC). In *Proceedings of the 14th International Workshop on Semantic Evaluation*.

Carlos S. Armendariz, Matthew Purver, Matej Ulčar, Senja Pollak, Nikola Ljubešić, and Mark Granroth-Wilding. 2020b. CoSimLex: A resource for evaluating graded word similarity in context. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 5878–5886, Marseille, France, May. European Language Resources Association.

Emily M Bender. 2013. Linguistic fundamentals for natural language processing: 100 essentials from morphology and syntax. *Synthesis lectures on human language technologies*, 6(3):1–184.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

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.

Shyam Boriah, Varun Chandola, and Vipin Kumar. 2008. Similarity measures for categorical data: A comparative evaluation. In *Proceedings of the 2008 SIAM international conference on data mining*, pages 243–254. SIAM.

Sung-Hyuk Cha. 2007. Comprehensive survey on distance/similarity measures between probability density functions. *City*, 1(2):1.

Christopher Cherpas. 1992. Natural language processing, pragmatics, and verbal behavior. *The Analysis of verbal behavior*, 10:135–47, 04.

Cedric De Boom, Steven Van Canneyt, Steven Bohez, Thomas Demeester, and Bart Dhoedt. 2015. Learning semantic similarity for very short texts. In *2015 ieee international conference on data mining workshop (icdmw)*, pages 1229–1234. IEEE.

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, June. Association for Computational Linguistics.

Christiane Fellbaum. 2012. Wordnet. *The encyclopedia of applied linguistics*.

Wael H Gomaa, Aly A Fahmy, et al. 2013. A survey of text similarity approaches. *International Journal of Computer Applications*, 68(13):13–18.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Hua He and Jimmy Lin. 2016. Pairwise word interaction modeling with deep neural networks for semantic similarity measurement. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 937–948.

Felix Hill, Roi Reichart, and Anna Korhonen. 2015. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Imgarylai. 2019. imgarylai/bert-embedding, Nov.

Derry Jatnika, Moch Arif Bijaksana, and Arie Ardiyanti Suryani. 2019. Word2vec model analysis for semantic similarities in english words. *Procedia Computer Science*, 157:160–167.

Thorsten Joachims. 1996. A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. Technical report, Carnegie-mellon univ pittsburgh pa dept of computer science.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Yoav Levine, Barak Lenz, Or Dagan, Dan Padnos, Or Sharir, Shai Shalev-Shwartz, Amnon Shashua, and Yoav Shoham. 2019. Sensebert: Driving some sense into bert. *arXiv preprint arXiv:1908.05646*.

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*.

George A Miller and Walter G Charles. 1991. Contextual correlates of semantic similarity. *Language and cognitive processes*, 6(1):1–28.

Jonas Mueller and Aditya Thyagarajan. 2016. Siamese recurrent architectures for learning sentence similarity. In *Thirtieth AAAI Conference on Artificial Intelligence*.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Yuval Pinter, Robert Guthrie, and Jacob Eisenstein. 2017. Mimicking word embeddings using subword rnns. *arXiv preprint arXiv:1707.06961*.

Didik Dwi Prasetya, Aji Prasetya Wibawa, and Tsukasa Hirashima. 2018. The performance of text similarity algorithms. *International Journal of Advances in Intelligent Informatics*, 4(1):63–69.

Gábor Recski, Eszter Iklódi, Katalin Pajkossy, and Andras Kornai. 2016. Measuring semantic similarity of words using concept networks. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 193–200.

Eric Sven Ristad and Peter N Yianilos. 1998. Learning string-edit distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(5):522–532.

Peter D Turney. 2008. A uniform approach to analogies, synonyms, antonyms, and associations. *arXiv preprint arXiv:0809.0124*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.