LIMSI-MULTISEM at the IJCAI SemDeep-5 WiC Challenge: Context Representations for Word Usage Similarity Estimation

Aina Garí Soler[1], **Marianna Apidianaki**[1,2] **and Alexandre Allauzen**[1]

[1]LIMSI, CNRS, Univ. Paris Sud, Université Paris-Saclay, F-91405 Orsay, France
[2]LLF, CNRS, Univ. Paris-Diderot
{aina.gari,marianna,allauzen}@limsi.fr

Abstract

We present the LIMSI-MULTISEM system submitted to the IJCAI-19 SemDeep-5 WiC challenge. The system measures word usage similarity in sentence pairs. We experiment with cosine similarities of word and sentence embeddings of different types, and with features based on in-context substitute annotations automatically assigned to WiC sentence pairs. The model with the highest performance on the WiC development set uses a combination of cosine similarities from different embedding types. It obtains an accuracy of 66.7 on the shared task test set and is ranked third among the participating systems.

1 Introduction

The SemDeep-5 WiC shared task proposes to identify the intended meaning of words in context. It is framed as a binary classification task that addresses whether two instances of a target word have the same meaning (Pilehvar and Camacho-Collados, 2019). The WiC dataset contains 7,466 sentence pairs and is proposed as a new evaluation benchmark for context-sensitive word representations.

We apply to this task the method from Garí Soler et al. (2019) which addresses the usage similarity of contextualized instances of words. The method integrates cosine similarities from different types of context-sensitive embeddings and in-context automatic substitutes. Our best system combines cosine similarities from three embedding types. It obtains an accuracy of 66.7 on the WiC test set, and is ranked third among all systems that participated in the task.

2 The WiC Dataset

The WiC dataset contains 7,466 sentence pairs of target words automatically labelled as having the same (T) or different (F) meaning. It was automatically compiled by extracting usage examples and sense information from lexical resources (WordNet (Fellbaum, 1998) VerbNet (Schuler, 2006) and Wiktionary[1]). To exclude instance pairs describing fine-grained sense distinctions, the resource was automatically pruned based on synset proximity in the WordNet network. Human accuracy upper bound on the dataset was defined as 80%, which corresponds to the average human accuracy on a sample of sentence pairs (Pilehvar and Camacho-Collados, 2019). Inter-annotator agreement was at the same level. The WiC dataset provides a benchmark for evaluating context-sensitive word representations, and their capacity to capture the dynamic aspects of word meaning and usage.

3 Contextualized Representations

Our proposed model computes a contextualized representation for each target word instance in a WiC sentence pair using different types of embeddings. The cosine similarity of the obtained vector representations is used as a feature for our classifier. We use the following types of embeddings:

SIF (Smooth Inverse Frequency): Simple method for deriving sentence representations from uncontextualized embeddings (Arora et al., 2017). Dimensionality reduction is applied to a weighted average of the vectors of words in a sentence. Weighting is based on word frequency in Common Crawl. We use SIF in combination with 300-*d* GloVe vectors trained on Common Crawl (Pennington et al., 2014).[2]

Context2vec: Neural model that learns embeddings for words and their contexts simultaneously (Melamud et al., 2016). It is based on word2vec's

[1]http://www.wiktionary.org/
[2]https://nlp.stanford.edu/projects/glove/

CBOW (Mikolov et al., 2013), but replaces the averaging of context word embeddings with a biLSTM that learns a representation of a sentence excluding the target word. We use a 600-*d* model pre-trained on the UkWac corpus (Baroni et al., 2009).[3]

ELMo (Embeddings from Language Models): Contextualized word representations obtained from the internal states of a deep bidirectional LSTM trained with a language model objective (Peters et al., 2018). Instead of learning the best linear combination of layer representations for a task – a common way of using ELMo – we use out-of-the-box 512-*d* embeddings.[4] We experiment with the top layer, and the average of the three hidden layers. We represent each WiC sentence in two ways: a) with the ELMo embedding corresponding to the target word, and b) with the average of ELMo embeddings of all words in the sentence. We also average the embeddings at a context window of size two, as this was shown to work better for word usage similarity with ELMo (Garí Soler et al., 2019).

BERT (Bidirectional Encoder Representations from Transformers): Representations obtained from a 12-layer bidirectional Transformer encoder trained with a language model objective where words on both sides of the target word in a sentence are masked and need to be predicted (Devlin et al., 2018). The pre-trained BERT architecture can be fine-tuned for specific tasks, but its internal contextualized word representations can also be used directly, similar to ELMo. We use 768-*d* uncased BERT representations of the target word, and the average of all words in a sentence.

USE (Universal Sentence Encoder): General-purpose sentence encoder trained with multi-task learning (Cer et al., 2018). Using transfer learning, USE improves performance on different NLP tasks at the sentence and phrase level (e.g. sentiment analysis). We use the Deep Averaging Network (DAN) encoder,[5] where input word and bigram embeddings are averaged and fed through a feedforward neural network, to create embeddings for WiC sentences.

[3] http://u.cs.biu.ac.il/~nlp/resources/downloads/context2vec/

[4] The medium-sized model at https://allennlp.org/elmo.

[5] https://tfhub.dev/google/universal-sentence-encoder/2

4 Automatic Substitution

Manual substitute annotations have been useful for in-context usage similarity estimation (Erk et al., 2009; McCarthy et al., 2016). The idea is that a high proportion of shared substitutes between two word instances reflects their semantic similarity.[6]

Extending previous work where manual substitute annotations were used to estimate usage similarity (Erk et al., 2009), we automatically annotate WiC instances with substitutes, and use features based on their overlap for our classifier. We use the context2vec method for automatic lexical substitution (Melamud et al., 2016). Given a sentence with a new instance of a target word t, and a set of candidate substitutes for the word ($S = s_1, s_2, ..., s_n$), context2vec ranks all candidates taking into account the target-to-substitute similarity and the substitute-to-context similarity.

$$c2v\_score = \frac{cos(s,t)+1}{2} \times \frac{cos(s,C)+1}{2} \quad (1)$$

In Formula 1, s and t are the context2vec word embeddings of a candidate substitute and the target, and C is the context vector of the sentence. The pool of candidate substitutes for a target word is formed from its set of paraphrases in the Paraphrase Database (PPDB) XXL package (Ganitkevitch et al., 2013; Pavlick et al., 2015).[7]

For every instance, context2vec ranks all candidates available for the target. Therefore, the generated ranking (R) always contains the same substitutes, in the same or different order. To make substitute overlap measures (McCarthy et al., 2016) operational in this setting, we use a filtering strategy from Garí Soler et al. (2019). The method detects a cut-off point in the ranking R that reflects a shift from good quality substitutes (high-ranked), to substitutes that are not a good fit in the context (low-ranked). It checks whether adjacent substitutes are paraphrases in PPDB; if not, it discards everything found after that point in R.

After filtering the ranking R for each sentence pair, we obtain three different features based on the retained substitutes.

- **Common substitutes:** The proportion of shared substitutes between the two instances of a target word.

[6] Previous work explores graded usage similarity, whereas in WiC it is binary.

[7] http://paraphrase.org/

Target	Sentences	Substitutes
way	Do you know the **way** to the airport?	ways, route, path, road {*connection, means, journey, move, direction, gateway, passage, place, ...*}
	He said he was looking for the **way** out.	ways, path, road, route, walk {*day, right, passage, move, means, time, doorway, ...*}
drink	Can I buy you a **drink**?	beer {*bottle, beverage, pint, vodka, booze, whisky, wine, liquor, drunk, cocktail, restaurant, ...*}
	He took a **drink** of his beer and smacked his lips.	swig {*bottle, pint, sip, drinking, beverage, drank, beer, drunk, cup, booze, liquor, ...*}

Table 1: Sentence pairs from the WiC training set for the noun *way* (gold label: T) and the verb *drink* (gold label: F) with automatic substitute annotations assigned by context2vec. Substitutes in italics were discarded after filtering.

- **GAP score:** GAP (Generalized Average Precision) considers the order of ranked elements and their weights (Kishida, 2005). GAP score ranges from 0 to 1 (for perfect disagreement/agreement). We take the average score between the rankings produced for a sentence pair in both directions ($GAP(R_1, R_2)$ and $GAP(R_2, R_1)$). Weights are the scores assigned to the substitutes by context2vec. We use the GAP implementation shared by Melamud et al. (2015).
- **Substitute cosine similarity**. We form pairs of substitutes from R_1 and R_2, and calculate the average of their GloVe cosine similarities. This feature accounts for the semantic similarity of substitutes, which can also, to some extent, reflect usage similarity.

A few WiC sentence pairs (5%) contain target words that are not present in the PPDB XXL package.[8] We apply automatic substitution to instances of target words that have paraphrases in PPDB, and back off to a classifier that uses only embedding-based features for the rest.[9] Table 1 shows examples of WiC sentences with automatic substitutes, before and after filtering.

5 Training Data Augmentation

We extend the WiC training data with 4,018 sentence pairs automatically extracted from the Concepts in Context (CoInCo) corpus (Kremer et al., 2014). CoInCo is a subset of the MASC corpus (Ide et al., 2008) which contains manual substitute annotations for all content words in a sentence. We use a balanced collection of similar (T) and dissimilar (F) sentence pairs from CoInCo, with labels automatically assigned based on substitute overlap (Garí Soler et al., 2019).[10] We apply the automatic substitution method described in Section 4, and extract substitute- and embedding-based features to be used by our models.

6 Model Development

We train a logistic regression classifier on the WiC training set, and experiment with different feature combinations on the development set. We use cosine similarities of different embedding representations. For ELMo and BERT, we try several layer combinations,[11] the target word vector and the sentence vector (see Section 3). For ELMo, we also apply a context window of size 2. The best configuration for BERT is the average of the last four layers, and for ELMo, the context window approach. We then combine the best embedding features for prediction. We also train models with the substitute-based features only, backing off to the best embedding-based model for instances of words not present in PPDB. We combine the best embedding- and substitute-based features in the Combined setting.

We apply the BERT and ELMO configurations that gave best results on the WiC development set to the setting with additional CoInCo data (WiC+CnC), and repeat the experiments. Results on the WiC development set are given in Table 2. Substitute-based features do not help the model,

[8] For full coverage, an option would be to use the whole vocabulary as a pool, as in the original context2vec implementation.

[9] PPDB paraphrases were available for target words in 97% of training, 89% of development and 90% of test sentence pairs in WiC.

[10] https://github.com/ainagari/coinco_usim_data/

[11] The average of the three layers or the top layer for ELMo. The top layer, the second-to-last layer, the average and the concatenation of the last four layers for BERT.

Features	WiC	WiC+CnC
BERT avg 4 tw	66.46	65.99
USE	63.64	63.48
ELMo top cw=2	62.38	61.76
SIF	60.66	59.56
c2v	60.34	61.13
BERT, USE	67.87	68.03
BERT, USE, ELMo	**68.65**	68.18
BERT, USE, ELMo, SIF	68.03	-
BERT, USE, ELMo, c2v	-	**68.34**
Substitute-based	60.34	57.84
Combined	66.77	68.34

Table 2: Accuracy of the models with embedding-based and substitute features on the WiC development set. We report results of the models trained only on WiC, and on the extended (WiC+CnC) dataset. The best configurations (marked in boldface) were applied to the WiC test set.

probably because of the noise in automatic annotations. The best result is obtained by the model trained only on WiC that uses cosine similarities from BERT, USE and ELMo. In the WiC+CnC setting, the Combined model gets the same performance as the model that uses four embedding types (BERT, USE, ELMo and c2v). We apply the simpler embedding-based model to the WiC test set.

7 Results and Analysis

Results of the two best-performing models (in boldface in Table 2) on the WiC test set are given in Table 3. Our best model is the one trained only on WiC, which uses BERT, USE and ELMo cosine similarities. It was ranked third at the competition with an accuracy of 66.71, which is higher than all results reported in the WiC description paper (Pilehvar and Camacho-Collados, 2019).

The additional training data extracted from CoInCo does not help the models. We believe this to be due to the different kind of sense distinctions present in the dataset extracted from CoInCo and in WiC. To explore this hypothesis, we take a closer look at the model predictions and carry out a qualitative analysis of the sense distinctions in the two datasets. The confusion matrices of the two best models on the development set show that wrong predictions most often concern dissimilar (F) sentence pairs. This type of error occurs more with the model trained on WiC+CnC (67% of total errors compared to 59% when training only on

Approach	Accuracy
WiC BERT, USE, ELMo	66.71
WiC+CnC BERT, USE, ELMo, c2v	65.64
BERT_{large} Threshold (Pilehvar and Camacho-Collados, 2019)	63.8

Table 3: Accuracy of our two best models on the WiC test set, compared to the best result from previous work.

WiC). A quick observation of WiC data reveals that dissimilar (F) pairs sometimes describe related senses, in spite of the pruning that aimed at excluding these from the dataset (Pilehvar and Camacho-Collados, 2019).

We extract a random sample of 60 sentence pairs from the CoInCo training data and the WiC development set to explore whether they differ in this respect. We manually annotate all pairs for graded usage similarity, using a scale of 1 (completely different) to 5 (the same), as in Erk et al. (2009). Our assumption is that F pairs that describe related senses will be assigned higher similarity scores. A comparison of the graded usage similarity values of gold F instances reveals that these values differ significantly in CoInCo and WiC ($p = 0.048$), as determined by a Mann-Whitney test, with WiC F pairs having a higher average similarity score (3.19 ± 1.52) than CoInCo F pairs (2.53 ± 0.19). The following F sentence pair from WiC is an example where the target word (*construction*) expresses different but closely related meanings (as a process and a result): *Construction is underway on the new bridge – The engineer marvelled at his construction*. The CoInCo sentence pairs extracted by Garí Soler et al. (2019) that we use for training describe more clear-cut sense distinctions, due to the process used for their extraction, based on the overlap of manually annotated substitutes (see Section 5).

8 Conclusion and Future Work

We propose a new model for word usage similarity estimation. The LIMSI-MULTISEM system combines different types of context-sensitive word and sentence representations with features derived from automatic substitution for usage similarity prediction. The best configuration combines cosine similarities from three embedding types: BERT, USE and ELMo.

In future work, we plan to use our model to investigate usage similarity on a per lemma basis, in

order to identify lemmas with clear-cut and fuzzy sense distinctions, as in McCarthy et al. (2016). This will help identify lemmas for which classification is trickier.

Acknowledgments

We would like to thank the anonymous reviewers for their helpful feedback. This work has been supported by the French National Research Agency under project ANR-16-CE33-0013.

References

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A Simple but Tough-to-Beat Baseline for Sentence Embeddings. In *International Conference on Learning Representations (ICLR)*, Toulon, France.

Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The WaCky wide web: a collection of very large linguistically processed web-crawled corpora. *Journal of Language Resources and Evaluation*, 43(3):209–226.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal Sentence Encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174, Brussels, Belgium. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805*.

Katrin Erk, Diana McCarthy, and Nicholas Gaylord. 2009. Investigations on word senses and word usages. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 10–18, Suntec, Singapore. Association for Computational Linguistics.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. Language, Speech, and Communication. MIT Press, Cambridge, MA.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The Paraphrase Database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 758–764, Atlanta, Georgia. Association for Computational Linguistics.

Aina Garí Soler, Marianna Apidianaki, and Alexandre Allauzen. 2019. Word usage similarity estimation with sentence representations and automatic substitutes. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics, Minneapolis, MN*. Association for Computational Linguistics.

Nancy Ide, Collin Baker, Christiane Fellbaum, Charles Fillmore, and Rebecca Passonneau. 2008. MASC: the Manually Annotated Sub-Corpus of American English. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Kazuaki Kishida. 2005. *Property of average precision and its generalization: An examination of evaluation indicator for information retrieval experiments*. Technical Report NII-2005-014E, National Institute of Informatics Tokyo, Japan.

Gerhard Kremer, Katrin Erk, Sebastian Padó, and Stefan Thater. 2014. What substitutes tell us - analysis of an "all-words" lexical substitution corpus. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 540–549, Gothenburg, Sweden. Association for Computational Linguistics.

Diana McCarthy, Marianna Apidianaki, and Katrin Erk. 2016. Word sense clustering and clusterability. *Computational Linguistics*, 42(2):245–275.

Oren Melamud, Jacob Goldberger, and Ido Dagan. 2016. context2vec: Learning Generic Context Embedding with Bidirectional LSTM. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 51–61, Berlin, Germany. Association for Computational Linguistics.

Oren Melamud, Omer Levy, and Ido Dagan. 2015. A Simple Word Embedding Model for Lexical Substitution. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 1–7, Denver, Colorado.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Proceedings of the International Conference on Learning Representations*, Scottsdale, Arizona.

Ellie Pavlick, Pushpendre Rastogi, Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2015. PPDB 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification . In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 425–430, Beijing, China. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word

representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2019. WiC: the Word-in-Context Dataset for Evaluating Context-Sensitive Meaning Representations. In *Proceedings of NAACL*, Minneapolis, United States.

Karin Kipper Schuler. 2006. *VerbNet: A Broad-Coverage, Comprehensive Verb Lexicon*. Ph.D. thesis, University of Pennsylvania.