MineriaUNAM at SemEval-2020 Task 3: Predicting Contextual Word Similarity Using a Centroid based Approach and Word Embeddings

Helena Gómez-Adorno[1] helena.gomez@iimas.unam.mx
Gemma Bel-Enguix[1] gbele@iingen.unam.mx
Jorge Reyes-Magaña[1,2] jorge.reyes@correo.uady.mx
Benjamín Moreno[3] bmm@xanum.uam.mx
Ramón Casillas[1] ramon.casillas@comunidad.unam.mx
Daniel Vargas[1] danizzvargas@gmail.com
[1]Universidad Nacional Autónoma de México, Ciudad de México, México
[2] Universidad Autónoma de Yucatán, Mérida, México
[3] Universidad Autónoma Metropolitana, Ciudad de México, México

Abstract

This paper presents our systems to solve Task 3 of Semeval-2020, which aims to predict the effect that context has on human perception of similarity of words. The task consists of two subtasks in English, Croatian, Finnish and Slovenian: (1) predicting the change of similarity, and (2) predicting the human scores of similarity, both of them for a pair of words within two different contexts. We tackled the problem by developing two systems, the first one uses a centroid approach and word vectors. The second one uses the ELMo language model, which is trained for each pair of words with the given context. Our approach achieved the highest score in subtask 2 for the English language.

1 Introduction

The aim of Semeval 2020 Task 3 (Armendariz et al., 2020), Graded Word Similarity in Context (GWSC), focuses on predicting the effect that context has in human perception of similarity of words.

The most common works in this area deal with the context where a word is placed to predict discrete changes in meaning: the different senses of a polysemous word (Castillo et al., 2008; Lossio-Ventura et al., 2016; Wang et al., 2018). However, this task highlights the context with more subtle (graded) effects on meaning, even for words not necessarily considered polysemous. That makes word similarity more challenging to be automatically predicted.

The main task is divided into two subtasks:

- Subtask 1: Predicting the change in the human annotator's scores of similarity when presented with the same pair of words within two different contexts. This task directly addresses our main question. It evaluates how well systems are able to model the effect that context has in human perception of similarity.

- Subtask 2: Predicting the human scores of similarity for a pair of words within two different contexts. This is a more traditional task which evaluates systems' ability to model both similarity of words and the effect that context has on it.

The datasets provided for this task contained contextual similarity ratings in four different languages: Croatian (HR), English (EN), Finnish (FI), and Slovenian (SL). The pairs of words come from the well-known SimLex999 (Hill et al., 2015) dataset.

The rest of the paper is organized as follows: in Section 2 our methodology is presented, along with some preliminary results using the corpus available in the practice phase. This will guide us to improve each approach. The final results of all systems with the evaluation corpus is reported in Section 3. The paper ends with some conclusions in Section 4.

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 150–157
Barcelona, Spain (Online), December 12, 2020.

2 Methodology

To approach the problem, we used two different methodologies. The first one, named A1, is inspired in a clustering algorithm that uses centroids to represent the center of a particular group of datapoints. The second one, named A2, uses ELMo embeddings (Peters et al., 2018) to represent words.

This task is not supervised, thus, we do not have a training corpus itself; instead, a practice kit was distributed by the organizers, composed by a reduced amount of word pairs with their given context, similarity, and similarity change score. It is important to notice that, for practical reasons, all the experiments were done with the English dataset, and the techniques were later extended to the other languages.

2.1 A1[1] - Basic Centroid

A Centroid is an artificial point in the space of records which represents an average location of a particular cluster(Khan and Mohamudally, 2010). Since we were working with words, we used different ways to transform them into numerical values. We considered the most common techniques in Natural Language Processing (NLP) to develop an algorithm that considered the following steps:

Pre-process the input data removing the punctuation symbols and stopwords of each context.

Vectorize the context words to arrays of numbers with the use of a term-document matrix. Each column correspond to a word in the vocabulary and each line correspond to a context (there are 2 contexts)

Calculate the similarity factor between contexts using the cosine similarity. The similarity matrix obtained for both contexts of the first example in the practice kit, is the following.

$$\begin{bmatrix} 1 & .1235 \\ .1235 & 1 \end{bmatrix} \tag{1}$$

This cosine similarity value is important to obtain the final results of the approach.

Transform the matrix using *tf_idf* (Ramos and others, 2003) to obtain the weight that each word has in the context.

$$\begin{bmatrix} 0 & 0.159 & 0.094 & 0 & 0 & 0.318... \\ 0.120 & 0.071 & 0.120 & 0.120 & 0 & 0.120... \end{bmatrix} \tag{2}$$

Obtain the centroid of each context in the matrix (2), computing an average, that is, to sum of the columns in the vector that represent the context and divide by the higher number of words without repetition. The results of this step are shown in the following matrix.

$$\begin{bmatrix} 0 & .07559 \\ 1 & .08189 \end{bmatrix} \tag{3}$$

Calculate the euclidean distance of each pair of words to the centroid of both contexts. The results of this calculation are shown in Table 1 corresponding to the words *manner* and *way*.

	manner	**way**
Context 1	.01844	.01215
Context 2	.01844	.01215

Table 1: Euclidean distance from words to centroids.

Considering the distances obtained, we established a decision rule in order to set which context corresponds to the pair of words and thereby, modifying the similarity factor of both considering the SimLex (Hill et al., 2015) original value and the cosine distance. The rule is: if both words have a smaller euclidean distance in context 1 compared with context 2, the cosine similarity will be added. The SimLex

Context 1	Context 2
7.4965	7.7435

Table 2: Similarity value using A1[1] for *manner* and *value*.

value of the pair *manner* and *way* is 7.62, and using this technique with the cosine value of .1235, we present the results in Table 2.

The results for all the samples using this approach are shown in Table 3. The columns *Context 1* and *Context 2* would give the outcome of subtask 2. In 6 of all the 8 samples, both words have a higher similarity when they are in context 2. Regarding the sample 3 and 6, the first context shows a higher similarity.

In order to resolve subtask 1, the value in context 2 is subtracted from the value in context 1. The results of this operation are shown in the column *Difference*.

Sample	Context 1	Context 2	Difference
1	7.49	**7.74**	0.247
2	7.55	**7.80**	0.246
3	**6.45**	6.30	-0.155
4	1.83	**2.16**	0.326
5	5.77	**6.02**	0.254
6	0.55	**0.80**	0.254
7	**5.02**	4.63	-0.390
8	2.52	**2.71**	-0.390

Table 3: Results using approach A1[1].

2.2 A1[2] - Embeddings Centroid

This approach is an evolution of the A1[1] described in section 2.1. We transformed the contexts into a matrix using pre-trained embeddings of each word. We used FastText (Bojanowski et al., 2017) embeddings [1] because they have models for all the languages involved in the competition.

For each of the two contexts of every sample, we built a matrix with dynamic dimensions that are adjusted automatically depending on the number of words in each case. The representation of each word is a vector of 300 dimensions; this value is given by FastText. Once each word is transformed in its corresponding vector, the centroid of each of the contexts is calculated. To compute the centroid, we average the vectors in the given context; i.e., we sum all word vectors and divide them by the total number of words in the context.

After this, we obtain the embedding representation of the pair of words and calculate the euclidean distances from each word to each context. The results obtained with this method can be observed in Table 4. With this, we can decide which of the contexts has a higher similarity for every word.

	manner	way
Centroid 1	2.74	2.27
Centroid 2	2.76	2.19

Table 4: Euclidean distance of *manner* and *way* to the centroids of each context.

The next step is to calculate the cosine distance between the two contexts using the centroids based on

[1] https://github.com/facebookresearch/fastText/blob/master/docs/crawl-vectors.md

embeddings. The results below were obtained with the first sample of the practice kit:

$$\begin{bmatrix} 1 & .8308 \\ .8308 & 1 \end{bmatrix} \tag{4}$$

We follow the same decision rule as before (see Section 2.1): We add the cosine similarity value between contexts to the SimLex original value between the pair of words and assign it to the context with higher Euclidean distance to the pair of words. This is because we want to increase the similarity of words in the context that is closer to such words. On the contrary, we subtract the cosine similarity value to the SimLex value between the pair of words and assign it to the context that has the lower euclidean distance to the words.

According to this decision rule, we obtain the values shown in Table 5 for the first pair of words in the practice kit. The results of the euclidean distance show that both words are more similar to context 2, so the cosine distance value is added to the SimLex value.

Context 1	**Context 2**
6.7891	8.4508

Table 5: Final similarity values of the words *manner* and *way* obtained with the algorithm based on Embeddings matrix.

Finally, Table 6 shows the results obtained for each sample of the practice kit with the algorithm based on embeddings matrix.

2.3 A1[3] - Embeddings Matrix with Multi Centroids

For this algorithm, instead of just adding the cosine similarity to the SimLex value of the pair of words, we have to decide which value will be added to or subtracted from the SimLex gold standard. So for this algorithm, we use the euclidean distance of each context to the pair of words and join this pair of words as a single cluster. The reason to join the words in a cluster is that, with a higher similarity value of words, we can have a new centroid that can be compared with the centroid contexts. With this action, we pretend to improve the performance of A1[2].
Now, we have three different centroids, namely CL_{C1} (Cluster context 1), CL_{C2} (Cluster context 2) and CL_{W} (Cluster for the words pair). The decision rule is the following:

1. if $Dist(CL_{C1}, CL_{W}) < Dist(CL_{C2}, CL_{W})$ then, the SimLex value for CL_{C1} is added to the $Dist(CL_{C1}, CL_{W})/10$ and the SimLex value for CL_{C2} is subtracted from the value of $Dist(CL_{C2}, CL_{W})/10$. This, will produce the predicted similarity values as follows:
   - $Sim(C1) = SimLex(WordsPair) + Dist(CL_{C1}, CL_{W})/10$
   - $Sim(C2) = SimLex(WordsPair) - Dist(CL_{C2}, CL_{W})/10$
2. Otherwise, if $Dist(CL_{C2}, CL_{W}) < Dist(CL_{C1}, CL_{W})$ then the SimLex value for CL_{C2} is added to the $Dist(CL_{C2}, CL_{W})/10$ and the SimLex value for CL_{C1} is subtracted from the value of $Dist(CL_{C1}, CL_{W})/10$. The final similarity values in this case, will be:
   - $Sim(C1) = SimLex(WordsPair) - Dist(CL_{C1}, CL_{W})/10$
   - $Sim(C2) = SimLex(WordsPair) + Dist(CL_{C2}, CL_{W})/10$

The results of this approach applied to the samples of the practice kit are shown in Table 7.

2.3.1 Results A1

Pearson and Spearman correlations are used to evaluate subtask 1 and subtask 2 respectively using the practice kit samples. In Table 8, we present all the results of the task.

So far, we can establish that the best performance is obtained with the Multi Centroid approach.

Sample	Context 1	Context 2	Difference
1	0.6789	**0.8451**	0.1661
2	0.6844	**0.8515**	0.1671
3	**0.7098**	0.5661	-0.1437
4	**0.2838**	0.1161	-0.1677
5	**0.6755**	0.5044	-0.1711
6	-0.1809	**1.5409**	1.7218
7	**0.5624**	0.4035	-0.1589
8	**0.3421**	0.1818	-0.1603

Table 6: Results with embeddings matrix.

Sample	Context 1	Context 2	Difference
1	0.7264	**0.7965**	0.0701
2	0.7108	**0.8264**	0.1156
3	**0.6876**	0.5862	-0.1014
4	**0.2420**	0.1548	-0.0872
5	**0.6366**	0.5425	-0.0941
6	0.0210	**0.1141**	0.0931
7	**0.5751**	0.5207	-0.0544
8	**0.3040**	0.2185	-0.855

Table 7: Results using embeddings matrix and multi centroids.

2.4 A2 - ELMo

This approach uses a contextualized word representation model that allows to capture variations in similarity between words, depending on the context. We used ELMo (Embeddings from Language Model) (Peters et al., 2018). Also, in order to have a baseline of context-independent words similarity, we used the Word2Vec algorithm (Mikolov et al., 2013). We select this model because Hill et al. (2015) mention that evaluations with this algorithm have the best correlation against the SimLex-999 gold-standard.

Finally, pre-trained datasets are used for both ELMo and Word2Vec, based on the fact that they are generally enough and compatible for both of the subtasks.

The model described in ELMo establishes that the highest level layers recover syntax characteristics, while the deepest layers preserve contextual characteristics. Therefore, in this approach, we use layer 2 vectors for the representation of words dependent on context. Likewise, we make a comparison of the vectors of layer 0, corresponding to the embeddings used for model training with Word2Vec pre-trained vectors.

The algorithm used in this approach to calculate the similarity between two words within a context is described as follows:

1. Lowercase and tokenize the sentence.

2. Calculate the vectors for each token, using the pre-trained ELMo model.

3. Recover the vector of the two words of interest in layer 2 of the ELMo model. $elmo\_l_2\_vector_{w_1}^{s_1}$ and $elmo\_l_2\_vector_{w_2}^{s_1}$, being *s* the context analyzed, in this case, sentence 1.

4. Calculate the cosine similarity between the two identified vectors, as follows:

$$ctx\_dep\_sim_{w_1:w_2}^{s_1} = cosine(elmo\_l_2\_{w_1}^{s_1}, elmo\_l_2\_{w_2}^{s_1})$$

	A1[1]	A1[2]	A1[3]
Pearson	-0.1146	0.399	0.5823
Spearman	0.9366	0.6455	0.952

Table 8: Pearson an Spearman correlations obtained for the practice kit.

2.4.1 Change between similarity scores for two different contexts

To visualize the similarity variation of two words in two different contexts, we can obtain the cosine similarity within each context and then calculate the difference using a subtraction. This will allow us to identify the existing variation. However, it is also useful to take a benchmark to know if that similarity increased or decreased. So, two base points are taken, the first one using layer 0 of the ELMo model, which represents the embeddings for entered tokens that were built through word count; and the second one, to Word2Vec embeddings, which have been generated taking into account the context during their training phase, but which are now invariant.
Then, we calculate the context-independent similarity for two given words w_1 and w_2, both of the ELMo layer 0 vectors, such as those belonging to Word2Vec.

$$elmo\_ind\_sim^{s_1}_{w_1:w_2} = 1 - cosine(elmo\_l_0\_{w_1}^{s_1}, elmo\_l_0\_{w_2}^{s_1})$$

$$w2v\_ind\_sim^{s_1}_{w_1:w_2} = 1 - cosine(w2v^{s_1}_{w_1}, w2v^{s_1}_{w_2})$$

Finally, we perform the subtraction between context-independent and context-dependent similarity, obtaining the difference between the similarities with respect to the benchmark for both the embeddings generated in ELMo layer 0 and Word2Vec:

$$\Delta elmo\_sim^{s_1}_{w_1:w_2} = ctx\_dep\_sim^{s_1}_{w_1:w_2} - elmo\_ind\_sim^{s_1}_{w_1:w_2}$$

$$\Delta w2v\_sim^{s_1}_{w_1:w_2} = ctx\_dep\_sim^{s_1}_{w_1:w_2} - w2v\_ind\_sim^{s_1}_{w_1:w_2}$$

If the difference is positive, it means that there was an increase in similarity between the two context-dependent words with respect to context-independent ones. Conversely, if the difference is negative, it would mean that the context-dependent similarity decreased with respect to the benchmark.

2.4.2 Results A2

We present the results corresponding to the words "task" and "woman", in the context-independent similarity for layer 0 of the ELMo and Word2Vec models we got -0.009 and 0.042 respectively.

Subsequently, we have the context-dependent similarity for each of the contexts, obtaining an increase in similarity in both contexts; in this case, 0.37 for the first sentence and 0.34 for the second. Finally, we calculate the similarity difference for each of the sentences, taking as a benchmark, first, the context-independent (layer 0) similarity of ELMo, giving an increase of 0.38 and 0.35 for the contexts 1 and 2 respectively. Next, we obtain the same difference between similarities, considering as a benchmark the context-independent similarities obtained with Word2Vec. In this case, there is also an increase in similarity of 0.33 for the first context and 0.30 for the second one. It is interesting to remark that these words, intuitively, seem to be unrelated. However, having them in context, their similarity increases notoriously, around 0.3 in both cases.

3 Results

We tested each approach in all the languages of the competition. Finally, we select our best scores regarding the evaluation phase, having different combinations for the two subtasks.

Approach	EN	HR	FI	SL
A1[1]	-0.105	-0.063	-0.353	0.038
A1[2]	0.052	-0.027	0.361	-0.166
A1[3]	0.029	-0.119	**0.389**	-0.130
A2	**0.544**	**0.374**	0.123	**0.328**

Table 9: Substask 1 results in evaluation phase. All languages.

Approach	EN	HR	FI	SL
A1[1]	0.686	**0.613**	0.537	**0.487**
A1[2]	0.477	0.256	0.079	0.202
A1[3]	**0.723**	0.602	**0.597**	0.479
A2	0.470	0.137	0.088	0.257

Table 10: Substask 2 results in evaluation phase. All languages.

Table 9 presents the scores obtained for subtask 1 in the evaluation phase. Considering our best results, we participated using the *ELMo* approach in English, Croatian and Slovenian. However, for the Finnish language we used the *Multi Centroid* approach.

Finally, we tested all the approaches in subtask 2, showing in Table 10 the results in all languages. The approaches selected were all based on Centroids, the *Multi* for English and Finnish, and the *Basic* for Croatian and Slovenian.

4 Conclusion

We present different approaches to solve the problem of predicting word similarity in different contexts.

The model introduced in A1, based on centroids, presents several iterative improvements that refine the method. Our team, **MineriaUNAM**, considers that centroid-based techniques are better to solve subtask 2.

The ranking positions we got in this competition are as follows: 1st place for English, 3rd in Croatian and Finnish, and 7th in Slovenian. The algorithm A1[3], based on multicentroids achieved the best correlation with human taggers on subtask 2.

The second approach, A2, is based on ELMo language model, which turns out to be better for subtask 1. The theoretical bases of the method seem to be correct, since using a context-dependent model guarantees a consistent association of the degree of similarity between two words within any given sentence. Besides, being a deep neural model, it is possible to work with a large amount of data, allowing this model to be general enough for tasks such as this one. However, this approach was unable to produce good similarity scores between the pair of words. The positions on the final board of substask 1 were as follows: 9th. in English and 8th. in Croatian, Slovenian and Finnish. In the centroid based approach, we used the SimLex original similarity values and modified it with respect to each context. In the case of the ELMo based approach the similarity is calculated independently from the SimLex. We believe that this fact, starting with a previous validated similarity score, was an important factor for the high performance of the centroid based approach.

Aknowledgements

This work has been supported by PAPIIT project IA401219, TA100520, and CONACYT project A1-S-27780.

References

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

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.

Carlos Castillo, Claudio Corsi, Debora Donato, Paolo Ferragina, and Aristides Gionis. 2008. Query-log mining for detecting polysemy and spam. In *Proceedings of the KDD Workshop on Web Mining and Web Usage Analysis (WEBKDD)*.

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

Dost Muhammad Khan and Nawaz Mohamudally. 2010. An agent oriented approach for implementation of the range method of initial centroids in k-means clustering data mining algorithm. *REASON*, 1(1):104.

Juan Antonio Lossio-Ventura, Clement Jonquet, Mathieu Roche, and Maguelonne Teisseire. 2016. Automatic biomedical term polysemy detection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1684–1688.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

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*.

Juan Ramos et al. 2003. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 133–142. Piscataway, NJ.

Wentao Wang, Nan Niu, Hui Liu, and Zhendong Niu. 2018. Enhancing automated requirements traceability by resolving polysemy. In *2018 IEEE 26th International Requirements Engineering Conference (RE)*, pages 40–51. IEEE.