Word Similarity Task

The results of the word similarity task are shown in Table 1, the Spearman and Pearson correlation values are similar, all of the values are positive, indicating that there is a positive correlation between the computed cosine similarities and the human judgement scores. It is notable that the BOW models work the best for the MEN dataset, whereas the dependency model gives the highest correlation for the SimLex dataset, due to an important difference: in SimLex, the words of a pair are of the same functionality, which is not always the case in the MEN dataset (e.g. dirty - washing). The word vectors in the dependency model capture the grammatical relations of words, which explains why the model works worse on the MEN dataset, where two words with different POS tags are compared.

To obtain the SimLex judgements, the participants had to rate the *similarity* (0-6) of the words of a pair. The MEN judgements were obtained differently: two word pairs were presented and the participant had to choose the pair of which the words were most *related*. This means that for the SimLex judgements, there were more possible values to assign to a word pair, meaning more variance in the scores, whereas for the MEN dataset there were only two. This might be a reason why the correlation values for the SimLex dataset are lower overall.

For the MEN dataset, the more related pair had to be chosen, which can also consist of words that are not similar in meaning but can occur in the same window-based context (e.g. airplane - flying). The BOW models are better at capturing the context, since they take the neighbouring words into account. The BOW5 model outperforms the BOW2 model, because it uses more context.

Word Analogy Task

Table 2 demonstrates that for the word analogy task, BOW2 performs best on both MRR and accuracy. Dependency is only slightly less accurate but significantly worse for MRR. This means that the BOW models ranks the correct words higher than dependency based model while the rank 1 words are roughly similar for both BOW2 and dependency.

The wrongly chosen words using BOW seem to be of similar meaning but with different syntactic relations than the target word, whereas dependency will have a similar form, but a different meaning. For example, "happy is to happily as precise is to ..." (precisely): BOW predicts *exact*, which is similar in context, and dependency predicts *flexibly*, since it will focus primarily on adverbs (here *-ly* words) in such cases.

Clustering Word Vectors

When looking at the 20 random sample words from each cluster, it becomes clear that the word clusters themselves have many clusters with just unclear relations but essentially BOW clusters yield more words that occur in the same window-based context, producing clusters that describe for example media, technology or nature. Dependency based clusters, however, perform better in clustering words based on their grammatical relations, such as names, locations, yet poorly in for example body parts. Additionally, BOW2 seems to have clearer cluster topics than BOW5 which might be due to too much context that is taken into consideration.

The number of clusters will have similar influence for both models. A small number of clusters (e.g. 5) will have a few good clusters, but also some bad ones because they are probably so general that there is no clear relation. More clusters (e.g. 15-20) yield many clear clusters. Using a lot of clusters (e.g. 50) causes an overfit which yields very few words in certain clusters and it causes similar topics to be split in an unusual manner.

Conclusion

For context based tasks, the BOW models will perform better since they capture the neighbouring words, whereas the dependency based model is more fit for tasks in which the relation between words and their functionality are important. Therefore, there is not one "best model" for all tasks, it really depends on the kind of task.

Appendix A Results

Dataset	Model	Spearman	Pearson
SimLex	**BOW 2**	0.4141	0.4285
	BOW 5	0.3674	0.3756
	Dependency	**0.4456**	**0.4619**
MEN	**BOW 2**	0.6999	0.6777
	BOW 5	**0.7232**	**0.7082**
	Dependency	0.6178	0.5974

Table 1: Word similarities for BOW2, BOW5 and dependency based models compared to the MEN and Word Similarity Task datasets using the Spearman and Pearson correlation efficients.

Model	MRR	Accuracy
BOW2	**0.7001**	**0.6218**
BOW5	0.6903	0.5966
Dependency	0.6764	0.6130

Table 2: Results of the word analogy tasks on the three models.

Appendix B Interpretation for Implementation

For the word analogy task, the exercise stated that vectors should be normalised (even given that the input was already normalised), however, since the comparisons of the vectors are done by cosine similarity, the lengths of the vectors should not make a difference (besides some float roundings). Our initial implementation returned the input word in the highest rank very often (e.g. "man is to woman as king is to ..." would return *king* instead of *queen*). Therefore the precision significantly lower while the MRR was just slightly lower than the shown results. The input words were thus removed from the ranking. To be sure about this, we both implemented the algorithm ourselves and compared this with the built-in `most_similar` function of `gensim`. The results did not differ and thus we conclude that normalising would not make a difference and that ignoring the input word was correct.

Regarding the clustering of word vectors, the exercise could be interpreted as that dimension reduction (to 2D) should be applied before the clustering is performed. This does not make sense since the complex clusters in the high dimensional space will not necessarily be close in the 2D space and therefore crucial information will get lost. Additionally, we did not put the clusters in this report due to the lack of space and lack of information it provides. Simply put, the clusters that are clustered in original space and transformed to 2D using PCA are (only) obvious using a few clusters (e.g. 5), but will not be clustered accordingly when the points in 2D space would be clustered since many clusters will seem to cross each other. Higher numbers of clusters are even less clear.

Appendix C Link to GitHub Repository

`https://github.com/meltjh/ull-1`