

Exploring the boundaries of vector space models for the WSI Task. CHERTOY system

Chernenko, Tatjana and Toyota, Utaemon

Institute for Computational Linguistics of University Heidelberg
Im Neuenheimer Feld 325, 69120 Heidelberg, Germany
<http://www.cl.uni-heidelberg.de/>

Formal Semantics course
WS 2017/2018
Dr. Vivi Nastase
Dr. Michael Herweg
March 30th, 2018

Abstract. In this paper we provide an approach to improve word sense induction systems (WSI) for web search result clustering. We experiment with language models, specific features, and clustering algorithms based on the sense2vec and the sent2vec systems. After having performed 40 carefully designed experiments we obtained interesting insights on the effects of several feature combinations which resulted in our WSI system CHERTOY.

Keywords: WSI · Web search result clustering · CHERTOY.

1 Introduction

With improved systems and devices and also people getting used to the quality and speed in information retrieval there is an increasing expectation in quality and speed for receiving - intentional or unintentional but proper - desired search results. Several methods provide approaches for solving clustering and diversifying web search results as well as solving query ambiguity issues. We present the CHERTOY unsupervised WSI system as our solution as a part of the project task of the Formal Semantics course of the Institute for Computational Linguistics of University Heidelberg in the winter semester 2017/2018. The task is based on the SemEval-2013 shared task [23]. For our experiments based on the sense2vec system we are using several language models, specific features, clustering algorithms, and their combinations to explore their effects and get an improvement in comparison to our baseline. We used pretrained language models as well as self-trained sent2vec models over the english Wikipedia from the 20th August 2017. Various preprocessing steps and compositional models are completing our setup. In order to get subtopic groups with our WSI systems, we grouped the web search result snippets over the topics provided as test data.

2 Related work

The complexity of the web and the amount of new content are still rising as the necessity for finding desired information. Even if the quality of web search results has become higher in the recent years there are still many problems to solve to achieve better and detailed results. A fundamental task is to solve ambiguous keywords in search queries [28]. Beside the word sense disambiguation (WSD) [22] the word sense induction (WSI) [24, 25] is an important approach for web search result clustering which was the aim for the SemEval task 2013 [23]. WSI aims at discovering senses of a given word from raw text based on unlabeled corpora [25]. There are several approaches for clustering search results based on context, words, phrases [4], co-occurrence graphs [12, 10] or probabilistic [5] methods.

To get similarities an algorithm needs to model and compare sentences [2, 3], phrases, words [17, 31] or characters. Algorithms such as *Siamese Continuous Bag of Words* averaging word embeddings [13], *ParagraphVector* which is learning fixed-length feature representations from variable-length pieces of texts [16] or the compositional n-Gram features using sent2vec [26] are providing attempts to sentence vector representations [32]. A distributional semantic model optimizing context predictions for phrases is *C-PHRASE* which can be used for single words as well as full sentences [14]. Vector space representation methods for words are the precise syntactic and semantic word relationship capturing *word2vec* or the global logbilinear regression model *GloVe* (Global Vectors) [27]. The character-enhanced word embedding model (*CWE*) is providing a method for languages such as Chinese or Japanese where a semantic meaning of a word is also related to the meanings of its composing characters [8]. Also for alphabetical based languages there are systems providing an model working with characters such as *Chargram* via character n-Grams [33].

Looking at subword information there are senses, hidden topics or context clues. An effort to build a model for senses provides the *sense2vec algorithm* [29]. Subword information using models are *Knet*, using contextual information and morphological word similarity to build a morphological knowledge based word embedding approach [9] or a system of Cao et al. using character n-grams, root/affix, inflexions and capturing structural information of their context with convolutional feature learning [6]. A model shared in the SemEval-2013 task is driven by a Latent Dirichlet Allocation (LDA) topic model [15]. Various unlabeled corpora as Wikipedia are used to feed the models with language and vocabulary cues [7]. Many algorithms produces vector spaces with a combination of several features and creating models to solve the WSI task. An overview over vector space models is provided by Turney et al. [30] and specially over vector representation of meanings in phrases and sentences by Mitchell et al. [21]. An objective evaluation of WSI systems is difficult differently to the evaluation of WSD systems but several frameworks are provided [1].

3 Model

The proposed CHERTOY system¹ is a quintessence of experiments within the confines of unsupervised WSI models that cluster the snippets into semantically-related groups, using word and sentence embeddings for language modeling. We compared our system against a simple baseline, based on sense2vec language model[29] and KMeans clustering algorithm with predefined number of clusters. In order to overcome the low results of the baseline, producing meaningless clusterings, we performed 40 experiments (see Tables 1-4 and Appendix), comparing the influence of different processing steps at the performance. CHERTOY is one of the experiments that achieves the highest average evaluation scores on the trial data (see Section 4). The main processing steps of CHERTOY are shown in Figure 1.

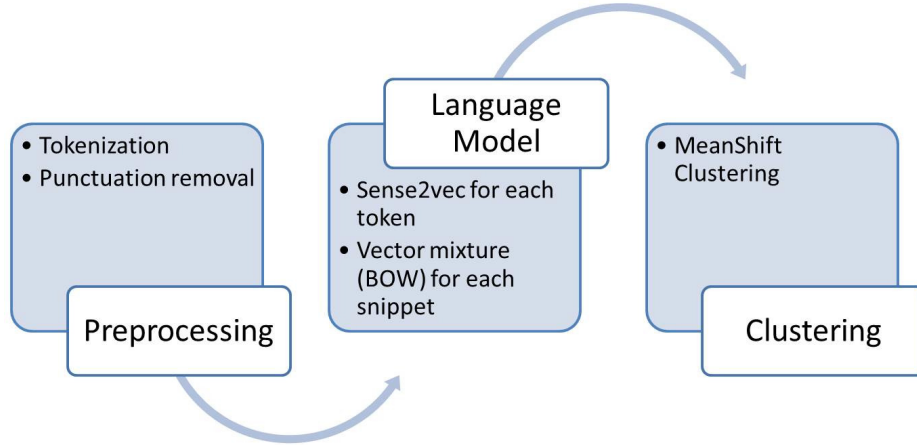


Fig. 1: Main processing steps of CHERTOY.

4 Discussion

SemEval 2013 task [11] provides a framework for the objective evaluation of Word Sense Induction algorithms in an end-user application [10]. We use the framework to monitor the system performance on the trial data with the changes that were made to the system. The framework provides four different evaluation measures: Average F1-measure, Average Rand Index, Average Adjusted Rand Index, and Average Jaccard Index.

¹ https://gitlab.cl.uni-heidelberg.de/semantik_project/wsi_chernenko_toyota.git

The Rand Index (RI) determines the percentage of snippet pairs that are in the same configuration in both clustering and gold standard clustering. Its main weakness is that it does not take chance into account. The Adjusted Rand Index (ARI) corrects this value and is 0 when the index equals its expected value [10]. However, the ARI measure tends to overweight the usefulness of snippets placed in different clusters. The Jaccard Index (JI) addresses this issue [10].

In this section, we consider the performance of the baseline and 40 experiments with it, and compare the impact of different features on the above-mentioned evaluation measures.

Table 1: Experiments - preprocessing.

Preprocessing steps
tokenization
tokenization + punctuation removal
tokenization + capitalization removal
tokenization + punctuation removal + en stopwords removal
tokenization + punctuation removal + stemming
tokenization + punctuation removal + POS
tokenization + punctuation removal + POS or without POS if no match
tokenization + punctuation removal + capitalization removal + english stopwords removal

Table 2: Experiments - language models.

Language models
sense2vec
word2vec (trained on Brown Corpus)
sent2vec with sent2vec_torontobooks_unigrams (2GB, trained on the BookCorpus dataset)
sent2vec wiki_bigrams
sent2vec wiki_unigrams
sent2vec own model, Wikipedia unigrams (25,4 GB, trained on Wikipedia 2017 dataset)
sent2vec own model, Wikipedia bigrams (36,6 GB, trained on Wikipedia 2017 dataset)

Table 3: Experiments - compositional models.

Compositional models
Vector Mixture Model (BOW model for each snippet)
tuned Vector Mixture Model (weighted BOW model for each snippet)

Table 4: Experiments - clustering.

Clustering
KMeans
MeanShift
Aglomerative Clustering
Spectral clustering
Affinity Propagation
cosine similarity

4.1 Baseline vs. CHERTOY

We show the results of the baseline and CHERTOY on the trial data in Table 5. CHERTOY significantly outperforms the baseline in pairwise evaluation metrics (av.RI, av.ARI, av.JI), showing almost equal to baseline F1 results.

Conceptually, the proposed system can be seen as an extension of the baseline, using additional preprocessing step of punctuation removal and exploring another clustering algorithm (MeanShift). These simple improvements double average Rand Index on the trial data and increase average Adjusted Rand index and Jaccard Index five-eight times over. Sections 4.2-4.4 prove that any other tested extensions within the confines of the described in Section 3 word/sentence embeddings models cannot reach the results of CHERTOY on the trial data, when we focus on pairwise evaluation measures. F1 measure ranks the systems differently, and some of our experiments outperform CHERTOY in F1 scores. For example, Variant 6 (see Appendix) reaches 0.7001 F1 score, clustering the snippets into 20 clusters, and Variant 16 (using Affinity Propagation clustering) reaches 0.6974 F1 score, which also slightly outperforms CHERTOY, but both of the systems show very low average RI, ARI, and JI values, producing trivial meaningless clustering. The slight variations of the CHERTOY system, represented by the Variants 37-40, utilize other language models (sent2vec[26]: two pretrained and two trained models (see Table 2)) instead of sense2vec[29]. In order to measure the effect of the quality of the language model on the performance, we used two given sent2vec models (Table 2) and trained our own uni- (25,4 GB) and bigrams models (36,6 GB). All these models are based on Wikipedia Corpus, therefore provide equal lexicon. As shown in Table 5, new language models influence F1, RI, ARI and JI measures, decreasing/increasing the values proportionally to the model size. At the same time, these variants cannot reach ARI score, performed by CHERTOY that uses sense2vec language model. It is noteworthy that all the variations of CHERTOY (Variants 37-40) have high Jaccard Index, which is not achievable by other tested systems (see Appendix). More specifically, CHERTOY benefits from the use of MeanShift Clustering algorithm in terms of JI performance. Our approach allows to reach state-of-the-art F1 and JI scores, using minimal processing steps and outperforming non-semantic SRC-systems, mentioned in [10]. As we don't test CHERTOY on the same datasets, the direct comparison is not possible. However, we can

expect comparable with state-of-the-art F1, RI and JI scores, performing worse than graph-based approaches in terms of ARI.

Table 5: CHERTOY vs. its variations and baseline.

Evaluation measure	Baseline	CHERTOY	Variant 37	Variant 38	Variant 39	Variant 40
average F1	0,6852	0,6832	0,6663	0,6739	0,6636	0,6636
average RI	0,3732	0,6823	0,6586	0,6582	0,6135	0,6093
average ARI	0,0284	0,1538	0,0064	0,0287	-0,0329	-0,0473
average JI	0,1431	0,6249	0,6496	0,6421	0,5963	0,5901
av. number of clusters	10	7	5	6	7,5	7,5
av. cluster sizes	10	14,82	21,9	17,56	13,54	13,54

4.2 Impact of preprocessing

With Variants 5, 7-10, and 13-15 (see Appendix) we explore the impact of preprocessing steps on the performance of the baseline with fixed processing steps and parameters (KMeans clustering with 7 clusters and sense2vec language model). The optimal results are attained after tokenization and punctuation removal (Variant 8), whereas an additional POS-tagging step can slightly improve RI, ARI and JI scores and reduce F1, possible because of the lower recall values. Other preprocessing steps (capitalization removal, stopwords removal, stemming and their combinations) slightly reduce all the evaluation measures.

4.3 Impact of different Language models

In order to measure the influence of different language models, we provide Variants 8, 11, 20, and 25-40 (see Appendix). Variant 11 that utilizes word2vec model, trained on Brown Corpus, slightly increases JI, but reduces F1, RI and ARI scores. Interesting results are achieved by using sent2vec toronto_books_unigrams model (Variant 20), which allows to improve RI, ARI and JI values (with a small decrease in F1). Other trained models - sent2vec wiki_unigrams and sent2vec wiki_bigrams - perform almost similar to sense2vec. Very good results are achieved using own trained unigrams and bigrams sent2vec models (25,4 GB and 36,6 GB respectively) with KMeans clustering, however sense2vec model provides higher ARI value (compare Variants 37-40 with CHERTOY; for more information see Section 4.1).

4.4 Impact of different Clustering algorithms

The interesting key finding is that, independently of the language model and preprocessing steps, the most noticeable affect is provided by the choice of the clustering algorithm. We consider five clustering algorithms: KMeans, Mean-Shift, Spectral Clustering, Agglomerative Clustering, and Affinity Propagation.

To check the performance of the algorithms, we also provide clustering with cosine similarity to the subtopics. Subtopics information is provided only for the trial data, thus Variants 21-23, 33-36 deal only for comparison with above-mentioned algorithms. The worst results are provided by Affinity Propagation algorithms, followed by KMeans and Agglomerative Clustering. Spectral Clustering allows to increase AJI, but shows negative ARI values. It shows the results comparable to those we could achieve, utilizing given subtopics and cosine similarity measure. MeanShift Clustering achieves the best performance, which is proved by testing it with different language models.

We have also explored the effect of the tuned vector mixture model and predefined number of clusters (to get further insights into the performance, see Appendix).

5 Conclusion

In this project, we explored the possibilities and confines of the word and sentence embeddings for the task of WSI [11]. This technique of language modeling is a simple but efficient method for Word Sense Induction (WSI), that is, methods aimed at automatically discovering the different meanings of a given term.

The main contribution of our work is a detailed research of the influence of different features, language models, clustering algorithms and their combinations at the performance of an unsupervised WSI system, using word and sentence embeddings. Our 40 experiments allow to choose the best combination of steps and to build a simple unsupervised WSI system CHERTOY that can identify the meaning of the input query and cluster the snippets into semantically-related groups. Our method shows performance improvements compared to the baseline and its derivatives.

References

1. Agirre, E., & Soroa, A. (2007, June). Semeval-2007 task 02: Evaluating word sense induction and discrimination systems. In *Proceedings of the 4th International Workshop on Semantic Evaluations* (pp. 7-12). Association for Computational Linguistics.
2. Agirre, E., Diab, M., Cer, D., & Gonzalez-Agirre, A. (2012, June). Semeval-2012 task 6: A pilot on semantic textual similarity. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation* (pp. 385-393). Association for Computational Linguistics.
3. Agirre, E., Cer, D., Diab, M., Gonzalez-Agirre, A., & Guo, W. (2013). * SEM 2013 shared task: Semantic textual similarity. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity* (Vol. 1, pp. 32-43).
4. Bernardini, A., Carpineto, C., & D'Amico, M. (2009, September). Full-subtopic retrieval with keyphrase-based search results clustering. In *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology-Volume 01* (pp. 206-213). IEEE Computer Society.

5. Brody, S., & Lapata, M. (2009, March). Bayesian word sense induction. In Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (pp. 103-111). Association for Computational Linguistics.
6. Cao, S., & Lu, W. (2017, February). Improving Word Embeddings with Convolutional Feature Learning and Subword Information. In AAAI (pp. 3144-3151).
7. Carmel, D., Roitman, H., & Zwerdling, N. (2009, July). Enhancing cluster labeling using wikipedia. In Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval (pp. 139-146). ACM.
8. Chen, X., Xu, L., Liu, Z., Sun, M., & Luan, H. B. (2015, July). Joint Learning of Character and Word Embeddings. In IJCAI (pp. 1236-1242).
9. Cui, Q., Gao, B., Bian, J., Qiu, S., Dai, H., & Liu, T. Y. (2015). Knet: A general framework for learning word embedding using morphological knowledge. *ACM Transactions on Information Systems (TOIS)*, 34(1), 4.
10. Di Marco, A., & Navigli, R. (2013). Clustering and diversifying web search results with graph-based word sense induction. *Computational Linguistics*, 39(3), 709-754.
11. Evaluating Word Sense Induction & Disambiguation within An End-User Application, <https://www.cs.york.ac.uk/semEval-2013/task11/index.php%3Fid=task-description.html>
12. Hope, D., & Keller, B. (2013, March). Maxmax: a graph-based soft clustering algorithm applied to word sense induction. In International Conference on Intelligent Text Processing and Computational Linguistics (pp. 368-381). Springer, Berlin, Heidelberg.
13. Kenter, T., Borisov, A., & de Rijke, M. (2016). Siamese cbow: Optimizing word embeddings for sentence representations. *arXiv preprint arXiv:1606.04640*.
14. Kruszewski, G., Lazaridou, A., & Baroni, M. (2015). Jointly optimizing word representations for lexical and sentential tasks with the c-phrase model. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers) (Vol. 1, pp. 971-981).
15. Lau, J. H., Cook, P., & Baldwin, T. (2013). unimelb: Topic modelling-based word sense induction. In Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013) (Vol. 2, pp. 307-311).
16. Le, Q., & Mikolov, T. (2014, January). Distributed representations of sentences and documents. In International Conference on Machine Learning (pp. 1188-1196).
17. Lin, D. (1998, August). Automatic retrieval and clustering of similar words. In Proceedings of the 17th international conference on Computational linguistics-Volume 2 (pp. 768-774). Association for Computational Linguistics.
18. Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
19. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111-3119).
20. Mikolov, T., Yih, W. T., & Zweig, G. (2013). Linguistic regularities in continuous space word representations. In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (pp. 746-751).
21. Mitchell, J., & Lapata, M. (2008). Vector-based models of semantic composition. *proceedings of ACL-08: HLT*, 236-244.

22. Navigli, R. (2009). Word sense disambiguation: A survey. *ACM Computing Surveys (CSUR)*, 41(2), 10.
23. Navigli, R., & Vannella, D. (2013). Semeval-2013 task 11: Word sense induction and disambiguation within an end-user application. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013) (Vol. 2, pp. 193-201)*.
24. Navigli, R., & Crisafulli, G. (2010, October). Inducing word senses to improve web search result clustering. In *Proceedings of the 2010 conference on empirical methods in natural language processing (pp. 116-126)*. Association for Computational Linguistics.
25. Navigli, R. (2012, January). A quick tour of word sense disambiguation, induction and related approaches. In *International Conference on Current Trends in Theory and Practice of Computer Science (pp. 115-129)*. Springer, Berlin, Heidelberg.
26. Pagliardini, M., Gupta, P., & Jaggi, M. (2017). Unsupervised learning of sentence embeddings using compositional n-gram features. *arXiv preprint arXiv:1703.02507*.
27. Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543)*.
28. Song, R., Luo, Z., Nie, J. Y., Yu, Y., & Hon, H. W. (2009). Identification of ambiguous queries in web search. *Information Processing & Management*, 45(2), 216-229.
29. Trask, A., Michalak, P., & Liu, J. (2015). sense2vec-A fast and accurate method for word sense disambiguation in neural word embeddings. *arXiv preprint arXiv:1511.06388*.
30. Turney, P. D., & Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37, 141-188.
31. Van Landeghem, J. (2016). A Survey of Word Embedding Literature. https://www.researchgate.net/profile/Jordy_Van_Landeghem/publication/301779119_A_Survey_of_Word_Embedding_Literature_Context_Representations_and_the_Challenge_of_Ambiguity/links/57279f7008ae262228b45180.pdf
32. Wieting, J., Bansal, M., Gimpel, K., & Livescu, K. (2015). Towards universal paraphrastic sentence embeddings. *arXiv preprint arXiv:1511.08198*.
33. Wieting, J., Bansal, M., Gimpel, K., & Livescu, K. (2016). Charagram: Embedding words and sentences via character n-grams. *arXiv preprint arXiv:1607.02789*.

Table 6: Performances

	Baseline	Variant 1	Variant 2	Variant 3	Variant 4	Variant 5
# of clusters for each topic	10	given	1	3	5	7
Preprocessing	tokenization					
Language model	sense2vec for words			tokenization sense2vec for words		
Compositional model	Vector mixture model (BOW (bag-of-words) representation with summarization for each snippet)					
Clustering	Kmeans					
Parameters	n_clusters=10	n_clusters=number of subtopics	n_clusters=1	n_clusters=3	n_clusters=5	n_clusters=7
Other						
Average F1	0,6852	0,6803	0,6610	0,6610	0,6656	0,6656
Average Rand Index	0,3732	0,3886	0,6928	0,4726	0,4156	0,3937
Average Adj. Rand Index	0,0284	0,0327	0,0000	0,0504	0,0303	0,0214
Average Jaccard Index	0,1431	0,1944	0,6928	0,3317	0,2347	0,1854
Average number of created clusters	10	7,5	1	3	5	7
Average cluster sizes	10,000	17,3611	100,0000	33,3333	20,0000	14,2857

	Variant 6	Variant 7	Variant 8	Variant 9	Variant 10	Variant 11
# of clusters for each topic	20	7	7	7	7	7
Preprocessing	tokenization	tokenization + capitalization removal	tokenization + punctuation removal	tokenization + punctuation removal + en stopwords removal	tokenization + punctuation removal + stemming	tokenization + punctuation removal
Language model			sense2vec for words			word2vec with Brown Corpus
Compositional model	Vector mixture model (BOW (bag-of-words) representation with summarization for each snippet)					
Clustering	Kmeans					
Parameters	n_clusters=20			clusters=7		
Other						
Average F1	0,7001	0,6705	0,6803	0,6656	0,6730	0,6750
Average Rand Index	0,0398	0,3937	0,4036	0,3910	0,3937	0,4023
Average Adj. Rand Index	0,0195	0,0212	0,0384	0,0224	0,0322	0,0006
Average Jaccard Index	0,0891	0,1876	0,1941	0,1876	0,1877	0,2059
Average number of created clusters	20	7	7	7	7	7
Average cluster sizes	5,0000	14,2857	14,2857	14,2857	14,2857	14,2857

Appendix

	Variant 12	Variant 13	Variant 14	Variant 15	Variant 16	Variant 17 "Chertoy"
# of clusters for each topic	20	7	7	7	7	7
Preprocessing	tokenization + punctuation removal	tokenization + punctuation removal + POS	tokenization + punctuation removal + POS or without sense2vec	tokenization + punctuation removal + capitalisation + en stopwords + POS	tokenization + punctuation removal	
Language model	sense2vec for words					
Tuned vector mixture model (BOW (bag-of-words) representation with						
Compositional model summarization for each snippet with $\alpha = 0.5$ for non-topic words and $\alpha = 1.0$ for topic words.)						
Clustering	Kmeans		Affinity		Propaga- MeanShift	
Parameters	clusters=7		default param			
Other						
Average F1	0,6852	0,6779	0,6663	0,6754	0,6974	0,6832
Average Rand Index	0,3958	0,4383	0,4344	0,3842	0,3691	0,6823
Average Adj. Rand Index	0,0223	0,0617	0,0693	0,0138	0,0299	0,1538
Average Jaccard Index	0,1818	0,2320	0,2407	0,1789	0,1313	0,6249
Average number of created clusters	7	7	7	7	14,25	7
Average cluster sizes	14,2857	14,2857	14,2857	14,2857	7,0753	14,8214

	Variant 18	Variant 19	Variant 20	Variant 21	Variant 22	Variant 23
# of clusters for each topic	default param	7	7	given	given	given
Preprocessing	tokenization + punctuation removal					
Language model	sense2vec for words	sent2vec for sentences with sense2vec.toronto books_unigrams model		sense2vec for words		
Compositional model	vector mixture model (BOW (bag-of-words) representation with summarization for each snippet)	vector mixture model (summarization of sentences for each snippet)	vector mixture model (BOW (bag-of-words) representation with summarization for each snippet)			
Clustering	Spectral ing	Cluster- Aglomerative Clustering	Kmeans	cosine similarity	cosine similarity with min=0.4	cosine similarity with min=0.3
Parameters	default param	n_clusters=7				
Other						
Average F1	0,6656	0,6710	0,6683	0,6610	0,6315	0,6614
Average Rand Index	0,6064	0,4192	0,4791	0,5333	0,4848	0,5309
Average Adj. Rand Index	-0,0391	0,0310	0,0727	-0,0052	-0,0006	-0,0086
Average Jaccard Index	0,5846	0,2170	0,3164	0,4891	0,4188	0,4861
Average number of created clusters	8	7	7	3,25	3	3,25
Average cluster sizes	12,5000	14,2857	14,2857	33,3333	31,3125	33,2083

Appendix

	Variant 24	Variant 25	Variant 26	Variant 27	Variant 28	Variant 29
# of clusters for each topic	20	7	7	7	7	7
Preprocessing	tokenization + punctuation removal			tokenization		tokenization + punctuation removal
Language model	sent2vec for sentences with sent2vec.toronto books-unigrams model	sent2vec (sent2vec, plain-text-unigramm - trained Model (Wikipedia 2017))	sent2vec (sent2vec, plain-texts_bigramm - trained Model (Wikipedia 2017))	sent2vec (sent2vec, sent2vec_senti2vec_wiki_bigrams)	sent2vec (sent2vec, plain-text_unigramm - trained Model (Wikipedia 2017))	sent2vec (sent2vec, plain-text_unigramm - trained Model (Wikipedia 2017))
Compositional model (summarization of sentences for each snippet)	vector mixture model					
Clustering	MeanShift					
Parameters	default param					
Other						
Average F1	0,6610	0,6823	0,6750	0,6610	0,6750	0,6896
Average Rand Index	0,6129	0,4222	0,4216	0,3913	0,3886	0,4714
Average Adj. Rand Index	0,0775	0,0477	0,0400	0,0191	0,0211	0,1003
Average Jaccard Index	0,6004	0,2343	0,2496	0,2221	0,2083	0,2815
Average number of created clusters	7	7,5	7,5	7,5	7,5	7
Average cluster sizes	14,4345	17,3611	17,3611	17,3611	17,3611	14,2857

	Variant 30	Variant 31	Variant 32	Variant 33	Variant 34	Variant 35
# of clusters for each topic	7	7	7	given	given	given
Preprocessing	tokenization + punctuation removal					
Language model	sent2vec (sent2vec, plain-texts_bigramm - trained Model (Wikipedia 2017))	sent2vec (sent2vec, sent2vec_wiki_bigrams)	sent2vec (sent2vec, plain-text_unigramm - trained Model (Wikipedia 2017))	sent2vec (sent2vec, plain-texts_bigramm - trained Model (Wikipedia 2017))	sent2vec (sent2vec, sent2vec_wiki_bigrams)	sent2vec (sent2vec, sent2vec_wiki_bigrams)
Compositional model	vector mixture model (BOW (bag-of-words) representation with summarization for each snippet))					
Clustering	Kmeans					
Parameters	n_clusters=7					
Other	Variant 8 + sentvec, plain-texts_bigramm - trained Model (Wikipedia 2017)	Variant 8 + sent2vec_wiki_bigrams	Variant 8 + sent2vec_wiki_unigrams	Variant 21 without POS + sim factor >0 + our sent2vec (sent2vec, plain-text_unigramm - trained Model (Wikipedia 2017))	Variant 21 without POS + sim factor >0, + sentvec, plain-texts_bigramm - trained Model (Wikipedia 2017)	Variant 21 without POS + sim factor >0, + POS + sim factor >0, + sent2vec_wiki_bigrams)
Average F1	0,6857	0,6703	0,6610	0,6610	0,6656	0,6679
Average Rand Index	0,4412	0,4102	0,4167	0,4531	0,5479	0,4071
Average Adj. Rand Index	0,0782	0,0057	0,0238	-0,0231	-0,0301	-0,0040
Average Jaccard Index	0,2598	0,2166	0,2458	0,3900	0,5312	0,2428
Average number of created clusters	7	7	7	3,75	2,25	6,75
Average cluster sizes	14,2857	14,2857	14,2857	30,0000	54,1667	18,3694

	Variant 36	Variant 37	Variant 38	Variant 39	Variant 40
# of clusters for each topic	given	default param	default param	default param	default param
Preprocessing	tokenization + punctuation removal				
Language model	sent2vec (sent2vec, plain-sent2vec_wiki_unigrams)	sent2vec (sent2vec, plain-text_unigramm - trained Model (Wikipedia 2017))	sent2vec (sent2vec, plain-texts_bigramm - trained Model (Wikipedia 2017))	sent2vec (sent2vec, sent2vec_wiki_bigrams)	sent2vec (sent2vec, sent2vec_wiki_unigrams)
Compositional model	vector mixture model (BOW (bag-of-words) representation with summarization for each snippet))				
Clustering	cosine similarity	MeanShift			
Parameters	n_clusters=given	default param			
Other	Variant 21 without POS + sim factor i 0, + sent2vec_wiki_unigrams	Variant 17 + our sent2vec (sent2vec, plain-text_unigramm - trained Model (Wikipedia 2017))	Variant 17 + sentvec, plain-texts_bigramm - trained Model (Wikipedia 2017)	Variant 17 + sent2vec_wiki_bigrams	Variant 17 + sent2vec_wiki_unigrams
Average F1	0,6533	0,6663	0,6739	0,6636	0,6636
Average Rand Index	0,4306	0,6586	0,6582	0,6135	0,6093
Average Adj. Rand Index	0,0390	0,0064	0,0287	-0,0329	-0,0473
Average Jaccard Index	0,2816	0,6496	0,6421	0,5963	0,5901
Average number of created clusters	7	5	6	7,5	7,5
Average cluster sizes	18,0106	21,9048	17,5595	13,5417	13,5417