APPENDIX D

Doc2Vec Word Embedding

Distributional hypothesis: the idea that words occurring in similar contexts have similar meanings [1] has provided the basis for a number of methods that use word co-occurrences in order

ALGORITHM 1: Similarity Function Algorithm for Character-Based Information Extraction

Input: Method calls in String format (c_1, c_2)

Output: Similarity Score

$Similarity\ (c_1, c_2)$ **begin**

if $(Length.c_1 < Length.c_2)$

then $Swap\ (c_1, c_2)$

BigLength $\longleftarrow Length.c_1$

Return (bigLength – *ComputeEditDistance*$^*(c_1, c_2)$) / bigLength

* *We have implemented the "Levenshtein distance" algorithm to compute the Edit distance*

to create vector representations of words (i.e. word embeddings), such that words with similar meaning have similar vectors [2, 3]. Le and Mikolov [4] proposed doc2vec as an extension to word2vec [5] to learn document-level embeddings. The goal of doc2vec is to create a numeric representation of a document, regardless of its length. After the model is trained, the word vectors are mapped into a vector space such that semantically similar words have similar vector representations (e.g., "strong" is close to "powerful") [4].

So when training the word vectors W, the document vector D is trained as well, and in the end of training, it holds a numeric representation of the document. While the word vectors represent the concept of a word, the document vector intends to represent the concept of a document.

ALGORITHM 2: Similarity Function Algorithm for Word-1 Method

Input: Two arrays of words (s_1, s_2)

Output: Similarity Score

$Similarity\ (s_1, s_2)$ **begin**

counter = 0

for each word w_1 **in** s_1

for each word w_2 **in** s_2

if $(w_1 == w_2)$

then counter = counter + 1

counter $\longleftarrow$ normalize* (counter)

Return (counter)

* *We normalized as: counter/(* $Length.s_1 \times Length.s_2$ *)*

TF-IDF

Term Frequency is the number of times a word has occurred in the document. TF-IDF scheme is used for extracting features or important words which can be the best representative of the document. It lowers the weight of the words that occur too often in all the sentences such as 'a', 'the', 'as' and increases the weight of the words that can be important in a sentence.

ALGORITHM 3: Similarity Function Algorithm for Word-2 Method

Input: Two arrays of Strings (s_1, s_2)

Output: Similarity Score

$Similarity\ (s_1, s_2)$ **begin**

raw_documents $\longleftarrow \{s_1, s_2\}$

for document d_i **in** raw documents

for document d_j **in** raw_documents

SimResult $\longleftarrow$ Similarity* (d_i, d_j)

Return SimResult

* *We used Doc2Vec algorithm*

TF-IDF is used to convert a document into structured format. The TF-IDF value increases proportionally to the number of times a word appears in the document, but is offset by the

ALGORITHM 4: Similarity Function Algorithm for Word-3 Method

Input: Two arrays of Strings (s_1, s_2)

Output: Similarity Score

$Similarity\ (s_1, s_2)$ **begin**

raw_documents $\longleftarrow \{s_1, s_2\}$

for word w_i **in** raw documents

for word w_j **in** raw_documents

SimResult $\longleftarrow$ Similarity* (w_i, w_j)

Return SimResult

* *We used TF-IDF algorithm*

frequency of the word in the corpus, which helps to control for the fact that some words are generally more common than others [6].

REFRENCES

[1] Z. S. Harris, "Distributional Structure," in *Papers in Structural and Transformational Linguistics*Dordrecht: Springer Netherlands, 1970, pp. 775-794.

[2] P. D. Turney and P. Pantel, "From Frequency to Meaning: Vector Space Models of Semantics," *Artificial Intelligence Research,* vol. 37, pp. 141-188, 2010.

[3] D. Kiela, F. Hill, and S. Clark, "Specializing Word Embeddings for Similarity or Relatedness," in *Empirical Methods in Natural Language Processing* Lisbon, Portugal, 2015, pp. 2044–2048: Association for Computational Linguistics.

[4] Q. Le and T. Mikolov, "Distributed Representations of Sentences and Documents," presented at the Proceedings of the 31st International Conference on Machine Learning, Proceedings of Machine Learning Research, 2014. Available: http://proceedings.mlr.press

[5] T. Mikolov, K. Chen, G. S. Corrado, J. Dean, and J. CoRR, "Efficient Estimation of Word Representations in Vector Space" 2013. ICLR

[6] S. CHIRANJIBI, "SEMANTIC TEXT CLUSTERING USING ENHANCED VECTOR SPACE MODEL NINGALI LANGUAGE," *COMPUTER SCIENCES AND TELECOMMUNICATIONS* vol. 4, no. 36, pp. 41-46, 2012. Scientific and Educational Organization "Internet Academy"