Statistical Natural Language Processing Project Draft

Anonymous submission

Paper ID ***

Abstract. The goal of our project in the course Statistical Natural Language Processing will be to develop a song lyric recommendation system for fan of the music artist. The algorithm works like described in the following sentences and visualized in Fig. 1 : Collect the song lyric words data of one artist and calculate embedded vectors by using embedding methods such as BERT or word2vec. As an input we get a keyword from the user, which is then like the songs emebedded into a euclidian space. With the keywords as an input, the model will return the song as well one line of the song, which is mostly related to the given keywords. The closeness of the input and the song lyric will be determined by a distance between input keyword vector and vectors of each one of words in lyric. For this project the key area of algorithm development will be finding similar documents, which not have to share words but topics. We therefore will evaluate our performance with two datasets. For the quantitative performance we will evaluate our performance on some news dataset, where news articles are classified according to some topics. [1]. For the qualitative performance, we will summerize a few songs manually and demonstrate that we get the same song suggest by our algorithm.

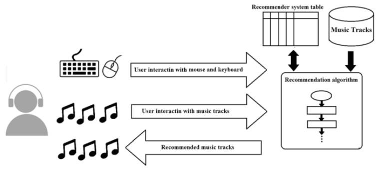

Fig. 1. Recommendation system workflow[2]

1 Related Works

1.1 Text representation

As the similarity measures are all operating in the Euclidean space, we will also discover, which methods are used to generate a mapping from text to the vector representation. In Figure 2 we visualize the structure of the different embedding methods.

One intuitive way to vectorize the text is by using one hot encoding

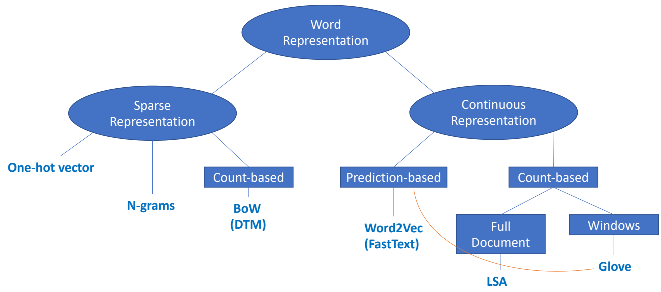

Fig. 2. Word Representation Techniques. As Glove can be categorized not only as a count based window method but also as a prediction method, it is connected to this category by an orange line

[3]. However, as their dimension is high, and the encoding does not reflect how often a word appears in a sentence, often different methods are used. A simple counting approach in embedding the documents into a Euclidean vector space is possible with Term Frequency (TF), which idea was first mentioned in [4]. This method however has the problem that it rewards, words, which often appeared in documents. This can lead to the problem that the documents are to become less indistinguishable. To focus more on the rare words, which make a text distinguishable, Term Frequency Inverse Document Frequency (TF-IDF) [5], [6], [7], [8] is used. To improve the effectiveness of TF and TF-IDF, the words get often redefined into n-gram groups [9]. Those groups are either formed by coupling of multiple words or characters. Depending on the task, the combined methods can then outperform the classical TF and TF-IDF approaches. A different count based approach is the Bag-Of-Words (BOW) method, where each sentence will be seen as a set of words and each duplicate will get removed [10]. The methods we discussed above are sparse representation methods and usually the result of these methods are relatively in highly dimension. Furthermore, with the vectors themselves, correlation information disappears. For example, in one-hot vector, correlation with the vectors are always 0. To reduce the dimensionality and see the correlation among the words better, continuous representation or distributed representation method suggested. One of the popular approaches is Latent Semantic Analysis (LSA), which is considered as an unsupervised learning approach. LSA [11] uses the statistics gathered from TF-IDF and then creates the embedding vectors for each document by reducing the rank of the TF-IDF matrix with a truncated SVD approach. Similar to LSA, GloVe [12] uses global matrix factorization as well to reduce the dimension of the data. The key difference here lies in the used data. GloVe uses local probabilities to calculate the global embedding vectors.

However, these methods do not provide any measures of semantic or lexical similarities [13]. One of the first models using a learning based approach to model also the probability function of the neighboring words is NNLM [14], which later got refined to RNNLM, with a recurrent architecture [15]. Another approach for distributed representation of a word embedding can be seen in [16]. The authors develop the hypothesis, that words, which share a syntactic context also tend to share a semantic meaning. One of the most popular embeddings implementing this hypothesis is word2vec [17]. Here, a Skip-gram neural network model predicts the surrounding words. The two main disadvantages of word2vec is the representation of rare words and the focus on local word structures. The first shortcoming is addressed in a followed up work, Fasttext [18]. Here are character ngrams are used to learn the word relations in contrast to word2vec, where they use the full word, with white spaces as a separator.

1.2 Deep Neural Network Models

With the advances in computational resources, neural network architecture gained popularity and their results improved drastically in the last 20 years, as network models got bigger and com There are two major design choices for Neural Networks. First, we have the feed-forward neural networks and secondly the recurrent neural networks (RNN). A specific and successful group of feed forward neural networks is a Convolutional Neural Network (CNN), with the break through paper [19]. It can be used for many text related tasked such as grammar correction [20]. However, one shortcoming of CNNs in NLP are the many convolutional layers. Passing through the multiple layers, the information gets compressed, which leads to a loss of sentence structure as well as information [21]. Here, different styles of frameworks have become popular in the last couple of years, which fix the issues of traditional CNN models and give better performance.
Often, RNN's can be used for improved performance. Here the Long short-term memory (LSTM) network designs [22] are worth noticing, as their memory capabilities help to prevent the information loss of CNN's. In [23] the authors shows that this network architecture, outperforms "comparable" CNN approaches in the domain of text classification. A popular LSTM network framework is ELMO [24]. This network design was one of the first using the bidirectional input measurements and achieving good scores. One of the most important neural network model BERT [25] uses the bidirectional transformer setup introduced in [26]. Over three years, BERT has become the benchmark for every new NLP model in regard to text embedding and next word prediction. Instead of using BERT, the USE (Universal Sentence Encoder) [27] is also a network design based on transformers. It can be however compared to BERT to register better differences in sentences, as it main task is: "semantic textual similarity (STS) between sentence pairs scored by Pearson correlation with human judgments" [27].

1.3 Similarity Metrics

On top of the traditional language evaluation method, perplexity [28], there are two different approaches to automatically evaluate language models based on comparing outcome with the ground truth which are *word based similarity metrics* and *word-embedding based similarity metrics* [29].
The metrics that evaluate the amount of word overlap between two texts are called Word overlap-based metric. Typically, BLEU [30], METEOR [31] scores which have been used for machine translation, and ROUGE [32] score that has been used for automatic summarization [29].
BLEU analyses the co-occurrences of n-grams in the ground truth and the proposed responses. METEOR [31] was introduced to deal with the weakness of BLEU. In contrast to BLUE, it also uses WordNet synonyms and stemmed tokens for the similarity analysis. Unlike BLEU and METEOR, ROUGE is Recall-oriented evaluation. It calculates a recall based score on the overlapping word occurrences. As an alternative to word-overlap based metrics, embedding based metrics are used [29]. The word-embeddings between the candidates and references are compared using a measure such as cosine distance. We will discuss three methods, Greedy Matching, Embedding average, and Vector extrema.
Greedy Matching [33], uses cosine similarity of token-level embedding, and averaging across all words to calculate the total score. This greedy approach favours responses with keywords that are semantically similar to those in the ground truth reference [29]. **Embedding Average** [34], [35] calculates sentence-level embedding using additive composition, a method for computing the meanings of phrases by averaging the vector representations of their constituent words. It is widely being used in textual similarity [29]. **Vector Extrema** [36] calculates sentence-level embedding and takes the most extreme value amongst all word vectors in the sentence. By taking the extrema along each dimension, we are thus more likely to ignore common words [29].

2 Project Plan

2.1 What do we want to do?

In our project, we will build a song recommendation system. This system suggest a song to the user, based on a keyword, provided by the user. Here the suggested song should be contentwise close to the given keyword or keyphrase. We therefore will embed the songs (further also called documents), into a euclidean space. There, the embedding method should generate clusters, which resemble the topics of the songs. We also require the embedding method to cluster "unseen" words into the right topics. Our algorithm now also embeds the keywords provided by the user into the same euclidean space. Now the document, which is measured to be the closest / most similar to the keyword, will be returned as the suggested song and also a summarization performed by an automatic tool will be created.

2.2 Data

To the best of our knowledge, there exist currently no dataset, which has lyrics and example keywords connected. We therefore will use two datasets. One for evaluating the quantitative performance of our algorithm, and a handcrafted lyric's dataset, which has all lyrics of an artist as well as some example keywords for each song. The first dataset is a labelled one, which already exists in [1]. It consists of news articles and their topics. We will use this labelled dataset to quantitative evaluate our method and optimize the performance and try out the most promising algorithms described in the literature review. In this dataset the labels and words, which are associated with the label, will be our generated keywords, and we want to "predict" news articles, which have the same label. This standardizes dataset allows us to create a baseline of our algorithm performance and also allows us to create a provable working version.
The second dataset will be used for qualitative purposes. With this dataset, after creating and thorough analysis of the data, we demonstrate the intended functionality of our algorithm. The dataset is based on a collection of songs from a specific artist. Here we will use a network like keyBERT to generate the related phrases for each song, and we will show, that our method can find a close matching song for each generated keyword. As a database for the lyrics, we will use the MLDB database [37].

2.3 Algorithms

The structure of the algorithm will be composed by two big steps, document embedding and summarization. The first step is document embedding. With the collected song dataset (MLDB), we will interpret one song as one document and embed those into the euclidean space with one of the embedding algorithms described in our literature review. Based on our literature review, we expect learning based distributed methods of word embeddings to perform the best, as they already have an understanding of language and are also able to embed words, which are not represented in our corpus. Therefore, similarity metrics based on cosine similarity (Embedding-based metric) will be the most promising ones to use for matching song and keyword. We will try out all three methods: Greedy Matching, Embedding Average, and Vector Extrema and evaluate on the news dataset, to find out which performs the best. After this, we summarize the song. With the embedded vectors for each song, we will run keyBERT network, which extract keywords out of documents. keyBERT is based on BERT[25] which was a state-of-the-art model in NLP tasks.

2.4 Evaluation Metrics

Our method can be broken down to a classification method of a matching between a keyword and a possible text within a class of documents. It is possible to evaluate the accuracy with a simple classification score, like the F_1 score [38]. This is only possible on the first dataset. However, on our self created dataset, we will demonstrate the functionality of our algorithm by qualitative experiments.

References

1. : Dataset for text association measurement. https://gofore.com/en/how-to-classify-text-in-100-languages-with-a-single-nlp-model/
2. Yousefian Jazi, S., Kaedi, M., Fatemi, A.: An emotion-aware music recommender system: Bridging the user's interaction and music recommendation. Multimedia Tools and Applications **80**(9) (2021) 13559–13574
3. Cohen, P., West, S.G., Aiken, L.S.: Applied multiple regression/correlation analysis for the behavioral sciences. Psychology press (2014)
4. Luhn, H.P.: A statistical approach to mechanized encoding and searching of literary information. IBM Journal of research and development **1**(4) (1957) 309–317
5. Dumais, S., Platt, J., Heckerman, D., Sahami, M.: Inductive learning algorithms and representations for text categorization. In: Proceedings of the seventh international conference on Information and knowledge management. (1998) 148–155
6. Joachims, T.: Text categorization with support vector machines: Learning with many relevant features. In: European conference on machine learning, Springer (1998) 137–142
7. Yang, Y., Liu, X.: A re-examination of text categorization methods. In: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. (1999) 42–49
8. Brank, J., Grobelnik, M., Milic-Frayling, N., Mladenic, D.: Interaction of feature selection methods and linear classification models. In: Workshop on Text Learning held at ICML, Citeseer (2002)
9. Vajjala, S., Banerjee, S.: A study of n-gram and embedding representations for native language identification. In: Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications. (2017) 240–248
10. Abdulaziz, W., M. Ameen, M., Ahmed, B.: An overview of bag of words;importance, implementation, applications, and challenges. (06 2019) 200–204
11. Landauer, T.K., Foltz, P.W., Laham, D.: An introduction to latent semantic analysis. Discourse processes **25**(2-3) (1998) 259–284
12. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). (2014) 1532–1543
13. Neto, J.L., Santos, A.D., Kaestner, C.A., Alexandre, N., Santos, D., et al.: Document clustering and text summarization. (2000)
14. Bengio, Y., Ducharme, R., Vincent, P.: A neural probabilistic language model. Advances in Neural Information Processing Systems **13** (2000)
15. Mikolov, T.: The rnnlm toolkit. In: RNNLM—recurrent neural network language modeling toolkit, IEEE Automatic Speech Recognition and Understanding Workshop. (2011)

16. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems **26** (2013)
17. Goldberg, Y., Levy, O.: word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. arXiv preprint arXiv:1402.3722 (2014)
18. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Transactions of the association for computational linguistics **5** (2017) 135–146
19. Chellapilla, K., Puri, S., Simard, P.: High performance convolutional neural networks for document processing. In: Tenth international workshop on frontiers in handwriting recognition, Suvisoft (2006)
20. Chollampatt, S., Ng, H.T.: A multilayer convolutional encoder-decoder neural network for grammatical error correction. In: Proceedings of the AAAI Conference on Artificial Intelligence. Volume 32. (2018)
21. Yao, H., Liu, H., Zhang, P.: A novel sentence similarity model with word embedding based on convolutional neural network. Concurrency and Computation: Practice and Experience **30**(23) (2018) e4415
22. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8) (1997) 1735–1780
23. Liu, P., Qiu, X., Huang, X.: Recurrent neural network for text classification with multi-task learning. arXiv preprint arXiv:1605.05101 (2016)
24. Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: 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), New Orleans, Louisiana, Association for Computational Linguistics (June 2018) 2227–2237
25. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
26. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
27. Cer, D., Yang, Y., Kong, S.y., Hua, N., Limtiaco, N., John, R.S., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., et al.: Universal sentence encoder. arXiv preprint arXiv:1803.11175 (2018)
28. Chen, S.F., Beeferman, D., Rosenfeld, R.: Evaluation metrics for language models. (1998)
29. Liu, C.W., Lowe, R., Serban, I.V., Noseworthy, M., Charlin, L., Pineau, J.: How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. arXiv preprint arXiv:1603.08023 (2016)
30. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th annual meeting of the Association for Computational Linguistics. (2002) 311–318

31. Banerjee, S., Lavie, A.: Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In: Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization. (2005) 65–72
32. Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. In: Text summarization branches out. (2004) 74–81
33. Rus, V., Lintean, M.: An optimal assessment of natural language student input using word-to-word similarity metrics. In: International Conference on Intelligent Tutoring Systems, Springer (2012) 675–676
34. Foltz, P.W., Kintsch, W., Landauer, T.K.: The measurement of textual coherence with latent semantic analysis. Discourse processes **25**(2-3) (1998) 285–307
35. Landauer, T.K., Dumais, S.T.: A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. Psychological review **104**(2) (1997) 211
36. Forgues, G., Pineau, J., Larchevêque, J.M., Tremblay, R.: Bootstrapping dialog systems with word embeddings. In: Nips, modern machine learning and natural language processing workshop. Volume 2. (2014) 168
37. : Website for the songs. http://www.mldb.org
38. Sokolova, M., Japkowicz, N., Szpakowicz, S.: Beyond accuracy, f-score and roc: a family of discriminant measures for performance evaluation. In: Australasian joint conference on artificial intelligence, Springer (2006) 1015–1021