VIREO @ TRECVID 2015: Video Hyperlinking (LNK)

Lei Pang and Chong-Wah Ngo

Video Retrieval Group (VIREO), City University of Hong Kong
http://vireo.cs.cityu.edu.hk

Abstract

This paper presents an overview and comparative analysis of our system designed for TRECVID 2015 [1] video hyperlinking (LNK) task. The application scenario for video hyperlinking is to satisfy the needs of users to find further information on the content of interest contained within an anchor. The task here is given an anchor to generate a ranked list of video segments relevant to the name entities extracted from the anchor. Our four runs are summarized below:

- ***tf-idf***: The subtitles of video segments are indexed with Lucene [2] and more like this (MLT) query is adopted for tf-idf based retrieval.
- ***word2vec***: Each video segment is represented by uniformly summing the vector representations of words in subtitles using Word2Vec [3, 4] and the relevant segments are ranked based on cosine similarity.
- ***weighted_word2vec***: The words in subtitles are weighted by document frequency and a video segment is represented by summing the weighted vector representations.
- ***linear_tf-idf_weighted_word2vec***: The relevant segments are ranked based on average fusion of *tf-idf* and *weighted_word2vec*.

1 System Overview

The dataset is composed of 3,518 BBC videos. The videos are accompanied by archival metadata (e.g., subtitles, list of popular UK celebrities) and automatic annotations (e.g., speech transcripts, shot segmentation, face detection, multiple versions of concept detectors). Among these rich information, we only explore the usage of ***subtitles*** to investigate the effectiveness of name entities for hyperlinking. The system consists of three stages: scene extraction based on topic detection (1.1), name entity detection (1.2) and hyperlinking with word vectors (1.3).

1.1 Scene Cutting

Since the dataset only provides "shot" segmentations and we actually want to work on "scene" level, we adopt TextTiling [5] to split the subtitles of videos into multi-paragraph subtopics, where each subtopic corresponds to a scene segment. The discourse cues for identifying major subtopic shifts are patterns of lexical co-occurrence and distribution. The algorithm has been proven to be useful for many text analysis tasks, including information retrieval and summarization [5]. We also constrain the algorithm to avoid

Table 1: P@N and MAP results for video hyperlinking

	P@5	P@10	P@20	MAP
word2vec	0.366	0.321	0.207	0.105
weighted_word2vec	0.338	0.344	0.227	0.126
tf-idf	**0.462**	0.406	0.300	0.180
linear_tf-idf_weighted_word2vec	0.436	**0.423**	**0.313**	**0.190**

splitting the consecutive speeches from the same speaker. Finally, a total of 100,917 scene segments are extracted.

1.2 Name Entity Detection

To facilitate the detection of hyperlinking targets, we extract name entities from the subtitles of each scene segment. Here, we adopt the Stanford Name Entity Recognizer (NER) [6] and a total of 98,601 distinctive name entities are detected. These name entities are classified into four categories – person, organization, location and others. We filter the noisy name entities based on document frequencies. The name entities with document frequency less than 10 are removed as noises and totally 8,168 name entities are retained. During indexing and retrieval, each name entity is treated as one single word.

1.3 Hyperlinking with Word Vectors

Word vector representation [3, 4] has shown great performance in measuring syntactic and semantic word similarities. Hence, different from the previous works, which usually enrich the text information with synonyms or conceptually connected words [7] or visual concepts [8], we directly represent each scene segment with word vector representation. As mentioned in [3], each scene segment is represented by summing all the words in the subtitle. The vector representation for each word is finetuned on the GoogleNews model[1]. The model contains 300-dimensional vectors for 3 million words and phrases. When finetuning, each word is initialized based on GoogleNews model and the name entities are initialized by summing the words. Since name entities are not very frequent, we adopt the skip-gram architecture with hierarchical softmax. Sub-sampling of the frequent words is also used for improving accuracy and speed.

In addition to uniformly summing all the word vectors as representation, we also want to measure the importance of different words. Here, we use document frequency (df) as weight. To further investigate the effectiveness of the vector representation, we further compare with tf-idf based retrieval and linearly combine the scores of these two measures.

2 Evaluation Results

Table 1 presents the evaluation results of our four runs, where P@N are precision-oriented metrics at different cutoff points and MAP is mean average precision. From the table, we can easily observe that the fusion of *tf-idf* and *weighted_word2vec* achieves the best performance. But it is surprise that *tf-idf* achieves better performance than both *word2vec* and *weighted_word2vec*. By observing the results, we find that *word2vec* and *weighted_word2vec* perform better when the anchors contains name entities, such as person names "James Humbert Craig" and locations "Cartley Hole". This is mainly due to the

[1] https://code.google.com/p/word2vec/

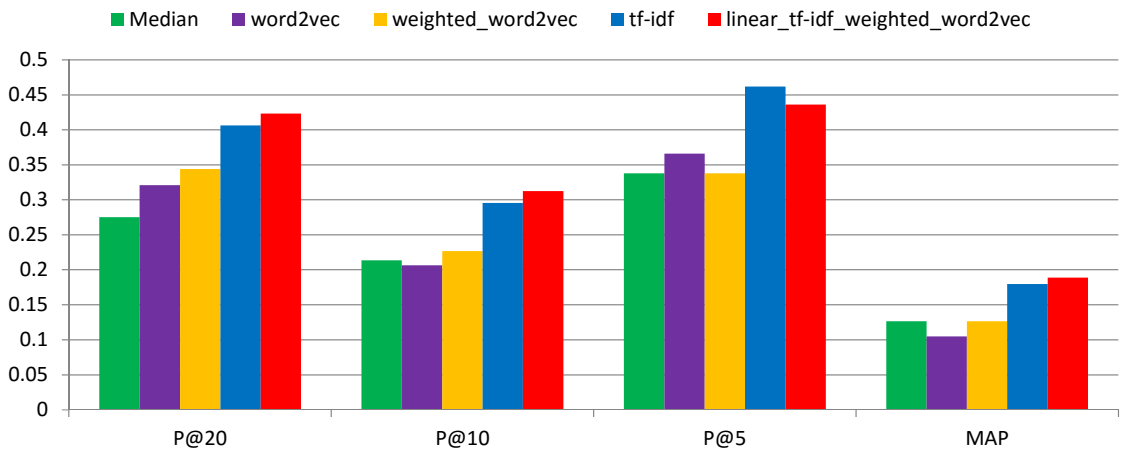

Figure 1: Results of four submitted runs in comparison with median perfromance

fact that word2vec can better represent the name entities by considering the context information. For example, the locations "Cartley Hole" is surrounded by words such as "house", "castle" and "garden" and the vector representation will be closely related to these words. However, *tf-idf* is attempting to retrieve the segments containing the exactly same entity, often resulting in lower recall. On the other hand, word vector representation also introduces semantic noises. For example, video segments about the traffic network is retrieved for the anchor showing tennis game the word "net" is mentioned. Since most of the anchors do not contain definite entities, the semantic noises degrade the performance of *word2vec* and *weighted_word2vec*. The linear combination achieving the best performance is in consistent with our observation, in which that the vector representation can be complementary to *tf-idf* by extending the keyword semantically.

We also compare our four runs with the median performance of other teams. As shown in Figure 1, *tf-idf* and *linear_tf-idf_weighted_word2vec* consistently performs better than median performance in all of the four measurements. Based on the previous observation, the performance of *word2vec* can be further improved through representing the segments with *interesting* words, such as nouns, verbs and name entities.

3 Summary

We submitted four runs mainly based on the word2vec representation. The word2vec indeed semantically extends the keywords for hyperlinking. However, some noises are introduced and as a result degrade the average performance. In the future, we are targeting to locate "interesting" words and representing the video segments based on the intensity of interest rather than document frequency. In addition, we will also consider visual entities such as faces and objects for hyperlinking.

4 Acknowledgement

The work described in this paper was supported by grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (CityU 11210514).

References

[1] P. Over, G. Awad, M. Michel, J. Fiscus, W. Kraaij, A. F. Smeaton, G. Quenot, and R. Ordelman, "Trecvid 2015 – an overview of the goals, tasks, data, evaluation mechanisms and metrics," in *Proceedings of TRECVID 2015*. NIST, USA, 2015.

[2] M. McCandless, E. Hatcher, and O. Gospodnetic, *Lucene in Action, Second Edition: Covers Apache Lucene 3.0*. Greenwich, CT, USA: Manning Publications Co., 2010.

[3] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *CoRR*, vol. abs/1301.3781, 2013.

[4] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems 26*, C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, Eds., 2013, pp. 3111–3119.

[5] M. A. Hearst, "Texttiling: Segmenting text into multi-paragraph subtopic passages," *Comput. Linguist.*, vol. 23, no. 1, pp. 33–64, 1997.

[6] J. R. Finkel, T. Grenager, and C. Manning, "Incorporating non-local information into information extraction systems by gibbs sampling," in *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, 2005, pp. 363–370.

[7] Z. Paroczi, B. Fodor, and G. Szücs, "Re-ranking the image search results for relevance and diversity in mediaeval 2014 challenge," in *Working Notes Proceedings of the MediaEval 2014 Workshop, Barcelona, Catalunya, Spain, October 16-17, 2014.*, 2014.

[8] B. Safadi, M. Sahuguet, and B. Huet, "When textual and visual information join forces for multimedia retrieval," in *International Conference on Multimedia Retrieval*, 2014, p. 265.