HULTECH at the NTCIR-11 IMine Task: Mining Intents with Continuous Vector Space Models

Jose G. Moreno
University of Normandie, France
GREYC, CNRS, UMR 6072, F-14032 Caen
jose.moreno@unicaen.fr

Gaël Dias
University of Normandie, France
GREYC, CNRS, UMR 6072, F-14032 Caen
gael.dias@unicaen.fr

ABSTRACT

In this paper, we present our participation in the Subtopic Mining subtask of the NTCIR-11 IMine task, for the English language. Our participation presents a novel strategy for intent mining given a list of candidates for a specific query topic. This strategy is based on a topic exploration through the use of continuous vector space models for each of the candidates based on classical vectorial operations. Our best run outperforms the other participants' submissions in terms of F-score and achieves a high position in the general ranking.

Team Name

HULTECH

Subtasks

Subtopic Mining (English)

Keywords

intent mining, continuous vector space models

1. INTRODUCTION

Adequate hierarchical intent identification for Web queries is a challenging task [2]. For this reason, the IMine Subtopic Mining subtask[1] for this year is concentrated on this challenge. In brief, this subtask consists in, given a query topic, returning as output a two level hierarchical structure of strings that cover the possible intents of the query. Moreover, the output must follow two restrictions: (1) the first level can only include a maximum of four different strings and (2) in the second level, a maximum of ten string could be associated to each string in the first level. For evaluation purposes, the organizers propose a set of novel metrics for hierarchical evaluation and others previously introduced in [9].

In this paper, we describe the HULTECH team participation in the mentioned challenge. Indeed, this is our second participation in the NTCIR tasks related to intent identification. In our previous participation [5], we propose a modified version of the classical k-means algorithm to identify intents through the clustering of search results [6]. An extended version of this strategy can be found in [7] where this strategy has been successfully tested in intent identification. For this year, we decide to focus our efforts in new exploratory ideas. Indeed, we try to introduce a novel technique, called continuous vector space models, to calculate word similarities instead of our previous word-word frequencies strategy.

Recently, many studies have been concerned with this task [8]. However, none of them have exploited recent advances in vectorial word representations. In this paper, we present a novel strategy that uses this new way to represent words with a special interest in the new available word-word operations. The underlying idea is that diversification of the intents can be achieved using classical vector operations. Results show that our strategy achieves top performance when compared to other participants of this task. In particular, our best run achieves the top position in terms of the F-score. Indeed, even when other participants outperform our method in terms H-measure, our method remains in the top three positions when evaluated with this metric. The remainder of this paper includes a brief introduction to continuous vector space models in Section 2, our intent mining strategy based on vectors in Section 3, results are presented in Section 4, and finally, discussion and conclusions are presented in Sections 5 and 6.

2. CONTINUOUS VECTOR SPACE MODELS

Recent ways to represent words are getting a lot of attention in NLP and IR communities. Indeed, the possibility of representing words with vectors is not new [1]. Some previous works have used decompositional methods to represent words as vectors [3]. But, some of the difficulties of these kind of studies are that they demand high computational power to achieve good performance in their representations. However, in [4] the authors have shown that adequate representations of words could be achieved with limited resources over large datasets. Their method relies in recursive and non-recursive neural networks. The idea is that continuous words could help to define the vectorial representation in a similar way that n-grams models can successfully define semantic relationships between words. Indeed, the use of neural networks is motivated by their strong capability to solve the underlying mathematical problem. When a certain level of convergence is achieved, the vectorial representation for each word is obtained from the hidden layer in the neural network. The good results obtained by this method as well as their new operation possibilities make it an interesting way to explore.

One of the most cited example is related with word relationships. In [4], it is shown that their model is capable

[1]Further information of the complete challenge and participants results could be found in [2].

	Position	H-score	F-score	S-score	H-measure
KUIDL-S-E-1A	1st	**0.9190**	0.5670	0.5964	0.5509
THUSAM-S-E-1A	2nd	0.8065	0.5179	0.4835	0.4257
HULTECH-S-E-2A	3rd	0.3596	**0.7184**	0.3977	0.1562
HULTECH-S-E-4A	4th	0.3055	0.6496	0.3981	0.1384
HULTECH-S-E-1A	10th	0.1703	0.7184	0.5754	0.0888
HULTECH-S-E-3A	11th	0.1703	0.7184	0.5754	0.0888

Table 1: Top three runs and our remaining submissions ordered by H-measure. In bold the best result (including others participants).

of discovering semantic relationships such as "Queen is to Woman as King is to Man". Their results strongly support this idea. Note that, if a vectorial representation is available, the vector of each word could be used to verify this situation. In the mentioned example, the case is validated due to the fact that over the full vocabulary, vector queen V_{queen} is the most similar vector to the resulting vector after a basic mathematical operation: $V_{king} - V_{man} + V_{woman}$. This method offers new ways to operate when comparison between words are needed. As far as we know, this paper is a first attempt to use these new kinds of representations in an intent mining experiment.

3. INTENT MINING USING CONTINUOUS VECTOR SPACE MODELS

In this work, we explore a continuous vector space models representation in the IMine Subtopic mining subtask. For that purpose, we utilize an existing vectors database of thousands of words built from a huge collection. This database is publicly available on the Web as well as the needed code to operate them[2]. An overall description of our algorithm is presented in Figure 1 and some of their intermediary phases are described below.

3.1 Mapping Subtopics to Vectors

In this phase, given a list of candidate subtopics we perform string matching to identify their vectorial representation for each subtopic. In that order, each string candidate is tokenized using a space as a separator and each token is searched in the vector database. If the token is found, then the vector is considered to represent the final string. When more than one token are found in the vector database the remaining vector is the sum along with their predecessors.

3.2 Intent Diversification with Vectors

The underlying idea consists in partial extractions of topics from the original query string. Using their respective vectorial representations, we first identify from a list of string candidates the most similar intent with respect to the query string. Note that this most similar candidate must cover the most reliable intent from the query. For the second intent, we subtract from the query the first intent and the most similar candidate to the difference is selected. Note that, the resulting vector from the subtraction expresses the remaining concepts not covered in the first intent and partially covered in the second intent. In an iterative process, the remaining intents are extracted using this subtraction strategy. Finally, the second level is built adding the most relevant string with respect to the first level.

[2]https://code.google.com/p/word2vec/ [Last access: 15/09/2014.].

3.3 Submitted runs

In order to evaluate our algorithm with different configurations, we submitted a total of four runs. The main differences are in the use of original candidate labels ($S_{candidates}$) or the modified version ($S'_{candidates}$). Another parameter that could be explored is the assigning method for the second level. We experiment with soft assigning and hard assigning. In the instance of the soft assigning, a total of ten strings are assigned to the second level of all the intents in the first level. In the case of hard assigning, a string in the second level is assigned to a unique intent in the first level. Our runs are identified as:

- HULTECH-S-E-1A: uses the modified version for first level and soft subtopics assigning.
- HULTECH-S-E-2A: uses the modified version for first level and hard subtopics assigning.
- HULTECH-S-E-3A: uses the original version for first level and soft subtopics assigning.
- HULTECH-S-E-4A: uses the original version for first level and hard subtopics assigning.

4. RESULTS

The complete information of the IMine results can be found in [2]. For evaluation a set of metrics are used including: H-score, F-score, S-score and finally a combination of them called H-measure. Indeed, H-score measures the quality of the hierarchical structure and evaluates if there is concordance between the second-level and the first-level, F-score measures the quality of the first-level subtopic and S-score measures the quality of the second-level subtopic. Finally, H-measure is defined as:

$$H\text{-measure} = H\text{-score} \times (\alpha \times F\text{-score} + \beta \times S\text{-score}) \quad (1)$$

where $\alpha + \beta$ is equal to one and both paraleters are fixed to 0.5. The top three performing runs (including all participants sorted by H-measure) and our other submissions are presented in Table 1. Note that, we get a higher performance in terms of F-score but it is not the same case when compared with the others participants in terms of H-score and S-score. A significant difference is clear between our best run (0.3596) and the top one (0.9190) in terms of H-score. This difference is not very relevant in terms of S-score.

Q a Web query string, $S_{candidates}$ the set of N strings candidates, K_{level1} the expected number of intents in level 1, K_{level2} the expected number of intents in level 2, $D = \{token_1, ..., token_m\}$ a dictionary of tokens and $V = \{vector_1, ..., vector_m\}$ the set of vectors that correspond to D.

Step 1 : Initialization V_Q.

The Q string is mapped to a vectorial representation.
A tokenization process is applied ($tokenization(Q) = \{Q_{t1}, ..., Q_{tq}\}$), then each token Q_{ti} is searched over D and their respective vector from V is extracted. Final vectorial representation of Q correspond to the sum of all the extracted vectors (V_Q).

Step 2 : Initialization $V_{candidates}$.

Each string from $S_{candidates}$ is mapped to a vectorial representation.
A tokenization process is applied ($tokenization(S^j_{candidates}) = \{S^j_{t1}, ..., S^j_{tq}\}$), then each token S^j_{ti} is searched over D and their respective vector from V is extracted. Final vectorial representation of $S^j_{candidates}$ correspond to the sum of all the extracted vectors ($V^j_{candidates}$). If any of the tokens if not found in D, a new string is defined extracting the non-found tokens from the original string ($S'_{candidates}$).

Step 3 : First level computation.

Define $S_{level1} = \emptyset$.
For l : 1 to K_{level1}, do:
Add to S_{level1} the associate $S'_{candidates}$ string to $V^j_{candidates}$ which is not included in S_{level1} that maximize: $Cosine(V_Q - \sum_1^{|V_{level1}|} V_{level1}, V^j_{candidates})$.

Step 4 : Second level computation.

For l : 1 to K_{level1}, do:
Define $S^l_{level2} = \emptyset$.
For m : 1 to K_{level2}, do:
Add to S^l_{level2} the associate $S_{candidates}$ string to $V^j_{candidates}$ which is not included in S^l_{level2} that maximize: $Cosine(V^l_{level1}, V^j_{candidates})$.

Return :

Finally, first level are the strings in S_{level1} and second level are the associated strings in S_{level2}.

Figure 1: The Vectorial based Intent Mining algorithm.

5. DISCUSSION

First, it is relevant to remark that our algorithm is constrained by the string candidates used as input. In our experiments, we have used only the query suggestions provided by the organizer. This string candidate set includes query suggestions from Bing, Google, Sogou, Yahoo! and Baidu. For many queries the number of candidates is around 20 and in some extreme cases the set of candidates only includes 16 strings[3]. Unfortunately, our experiments were limited by these candidates, so that clearly explains our under performance in terms of H-score that penalize the inconsistent relations between the first and second level. In future work, we plan to include other string candidates to increase the number of intents in the second level when the hard assigning is used. In that case the performance is better than with soft assigning, but it is not as good when compared with the other participants. This situation is due to the low number of candidates. In many cases our second level submission includes less than the maximum of ten intents allowed for the task.

6. CONCLUSIONS

This paper presents a novel strategy for intent mining and diversification using continuous vector space models. Results show that our proposal outperforms others participants in terms of F-score and achieves the third position when compared in terms of H-measure. We propose the integration of more adequate string candidates for the mining task in order to improve the obtained performance in terms of S-score.

[3]This situation occurs with the query id 56.

7. REFERENCES

[1] G. E. Hinton. Distributed representations. *Parallel distributed processing: Explorations in the microstructure of cognition*, 1, 1984.

[2] Y. Liu, R. Song, M. Zhang, Z. Dou, T. Yamamoto, M. Kato, H. Ohshima, and K. Zhou. Overview of the ntcir-11 imine task. In *NTCIR-11*, 2014.

[3] K. Lund and C. Burgess. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, & Computers*, 28(2):203–208, 1996.

[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*, pages 3111–3119, 2013.

[5] J. G. Moreno and G. Dias. Hultech at the ntcir-10 intent-2 task: Discovering user intents through search results clustering. In *NTCIR-11*, 2013.

[6] J. G. Moreno, G. Dias, and G. Cleuziou. Post-retrieval clustering using third-order similarity measures. In *51st Annual Meeting of the Association for Computational*

Linguistics (ACL), pages 153–158, 2013.

[7] J. G. Moreno, G. Dias, and G. Cleuziou. Query log driven web search results clustering. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR)*, pages 777–786, 2014.

[8] T. Sakai, Z. Dou, T. Yamamoto, Y. Liu, M. Zhang, M. Kato, R. Song, and M. Iwata. Summary of the ntcir-10 intent-2 task: Subtopic mining and search result diversification. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 761–764, 2013.

[9] T. Sakai and R. Song. Evaluating diversified search results using per-intent graded relevance. In *Proceedings of the 34th international ACM conference on Research and development in Information Retrieval (SIGIR)*, pages 1043–1052, 2011.