Individual project. Randomised algorithms

Short text understanding

Konstantin Frolov, Innopolis University

I. Introduction

THIS work represents the implementation of the algorithm proposed in [1]. The purpose of the analysed algorithm is machine understanding of short texts (about 10 words long) like web-search queries, micro-blogs, advertisements etc. The following peculiarities of short texts are taken into account:

- Necessity to process instantly and in real time
- Highly probable inconsistency with regular grammar and language semantics
- Limited context

Based on the reasons mentioned above, the algorithm comprises online and offline stages. The latter is aimed at offline knowledge acquisition. This is quite an important step whose results are used at all stages of the online part. The online part is divided into three stages:

1) Text segmentation
2) Type detection
3) Concept labelling

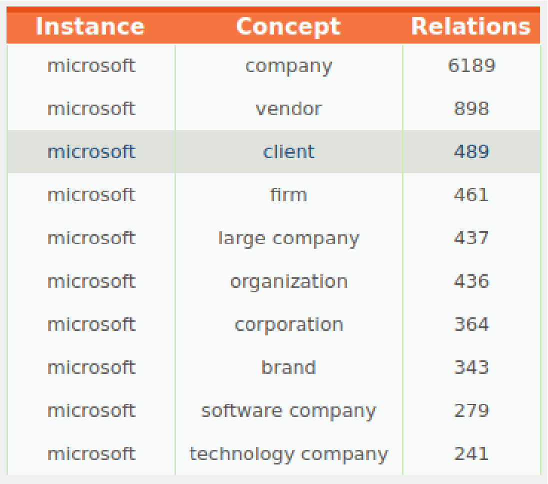
In this work, the text segmentation part is implemented along with the corresponding offline portion.

II. Problem statement

Due to the nature of the types of short texts described in section I, traditional methods of Natural Language Processing are inefficient at the task. One of the main reasons is that the text might not be entirely grammatically correct. Secondly, the word order cannot usually be relied upon in the same way as in ordinary texts. Additionally, some unimportant words like prepositions may be omitted. Thirdly, the context is tightly limited.

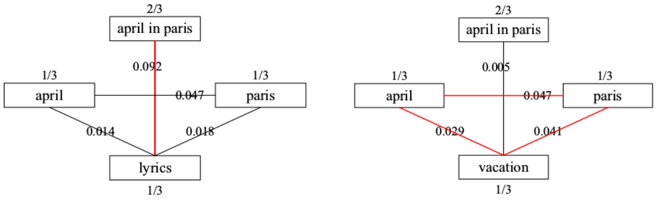
III. Model description

The challenges mentioned in the previous section are targeted by several measures. As for the shortage of context data, a substantial data set[1] containing terms with their different meanings and frequencies of them collected over the Internet is used (see Figure 1). The proper labelling of text terms and segmentation of the text into logical units are achieved by identifying the sequence of concepts corresponding to the original terms which has the highest weight. The weights of candidate sequences are calculated with the information contained in the data set.



Instance	Concept	Relations
microsoft	company	6189
microsoft	vendor	898
microsoft	client	489
microsoft	firm	461
microsoft	large company	437
microsoft	organization	436
microsoft	corporation	364
microsoft	brand	343
microsoft	software company	279
microsoft	technology company	241

Fig. 1: Section from labelled instances base



(a) Segmentation of ”april in paris”, option 1

(b) Segmentation of ”april in paris”, option 2

Fig. 2: Segmentation options of sample sentence

As it was mentioned above, the first step of the online stage is text segmentation. The process and the importance of it can be demonstrated utilising two search query examples: ”April in Paris lyrics” and ”vacation April in Paris” (see Figure 2). It is clear that in the former, as a song title, the ”April in Paris” term must be treated as a single one, whereas in the latter, the meaning of each word can only be identified correctly when it is split.

Text segmentation is carried out in two steps. At the first step, all possible splittings are added to the graph. In this graph, only those nodes are linked, which belong to the same sequence spanning the whole sentence, thus each sequence forming a clique in the graph. At the second step, a clique with maximum average edge weight is selected from the graph. The weights are assigned to edges by taking a cosine product of two terms:

$$w = cosine(\vec{x}, \vec{y})$$

$\vec{x}$, $\vec{y}$ - concept vectors of two terms obtained from the knowledge base.

The maximum clique is calculated using a randomised algorithm. The first step is to select a random edge with a probability in proportion to its weight. Next, all vertices

[1] Probase data publicly available at http://probase.msra.cn/dataset.aspx

not connected to the edge’s nodes are removed from the graph along with all corresponding nodes. This procedure is repeated, gradually reducing the number of candidates, until only one clique remains. Although the probability of selecting an edge depends on its weight, thus making the largest-weight candidate clique more likely to be produced, errors are possible due to the randomness of the algorithm. To overcome this, the authors of the original paper [1] recommend to run it several times and select the best result. The advantage of such an approach is the increase of speed of overall text processing which is essential in such sort of problems.

IV. Experiment

The work of the algorithm was analysed on the following phrases:

TABLE I: Experiment results

Original	**Segmented**
April in Paris lyrics	’april’, ’paris’, ’lyrics’
April in Paris vacation	’vacation’, ’april in paris’
Hotel California Eagles	’eagles’, ’hotel california’
Read Harry Potter	’read’, ’harry’, ’potter’
Read Harry Potter book	’read’, ’book’, ’harry potter’
Watch Harry Potter movie	’watch’, ’movie’, ’harry potter’
Manchester City beat Manchester United and won the trophy	’manchester’, ’city’, ’beat’, ’trophy’, ’manchester united’
Niagara falls best season to visit	’best’, ’season’, ’visit’, ’niagara falls’
How to hone randomized algorithms	’hone’, ’randomized’, ’algorithms’, ’how to’

V. Conclusion

Table I demonstrates that the algorithm makes crucial errors in some phrases. One of them is ”April in Paris lyrics”. This error can be explained by the fact that a simplified method of calculation of edge weights was used in this work. Originally, the weight also takes into account the variety of parts of speech which can be related to a term, and it is also based on the cosine product of term concept cluster vectors instead of plain concept vectors.

Another problem to overcome is low speed of weights calculation. A method must be found of calculation of cosine product of entries contained in a table whose entries number in the tens of millions.

References

[1] W. Hua, Z. Wang, H. Wang, K. Zheng and, X. Zhou, ”Short Text Understanding Through Lexical-Semantic Analysis”, *Data Engineering (ICDE) Conf.*, Seoul, South Korea, April, 2015.