

Adaptive Context-based term (re)weighting

An experiment on Single-Word Question Answering

Marco Ernandes and Giovanni Angelini and Marco Gori and Leonardo Rigutini and Franco Scarselli¹

Abstract. Term weighting is a crucial task in many Information Retrieval applications. Common approaches are based either on statistical or on natural language analysis. In this paper, we present a new algorithm that capitalizes from the advantages of both the strategies. In the proposed method, the weights are computed by a parametric function, called *Context Function*, that models the semantic influence exercised amongst the terms. The Context Function is learned by examples, so that its implementation is mostly automatic. The algorithm was successfully tested on a data set of crossword clues, which represent a case of Single-Word Question Answering.

1 Introduction

Term weighting is an important task in many areas of Text Processing, including, Document Retrieval, Text Categorization and Question Answering (QA). The goal of term weighting is to assign to each term w found in a collection of text documents a specific score $s(w)$ that measures the importance, with respect to a certain goal, of the information represented by the word. Common approaches to term weighting can be divided into two groups: statistical and linguistic techniques. Statistical techniques [1] (e.g. TFIDF) are efficient and easy to develop, but they tend to consider the words of a document as unordered and independent. The techniques inspired by natural language theories [6], as morphological analysis, naturally exploit the information provided by word contexts. This makes the processing more expressive, but also slower and more complex to design.

In this paper, we present a term weighting algorithm that aims to combine the advantages of both statistical and linguistic strategies. The method exploits the relationships among the words of a document. The intuition is that the relevance of a term can be computed recursively as the combination of its intrinsic relevance and the relevance of the terms that appear within the same context. The influence exercised by a word on another one is computed using a parametric function, called *Context Function*. This function can use both statistical and linguistic information, and it can be trained by examples.

The Context-based algorithm has been evaluated on a specific problem, that of Single-Word Question Answering (QA), where the goal is to find the single correct word that answers a given question. The experimental results prove that the approach is viable.

2 Adaptive Context-based term (re)weighting

The proposed method exploits the word contexts. A word context primarily consists of the text surrounding a given word, but could also include other features, e.g. document titles, hyper-linked documents. The basic idea is that, in order to measure the relevance of a word with respect to a certain goal (e.g. a query, a document category), the features of the context in which the term appears are important as

well as the features of the word itself. In this work we assume that a text document can be represented by a social network [5], where the importance of the words can be computed on the basis of their neighbours. More precisely, the weight $s(w)$ of the word w is computed as

$$s(w) = (1 - \lambda)d_w + \lambda \sum_{u \in r(w)} s(u)c_{w,u}, \quad (1)$$

where d_w is the *default score* of w , $r(w)$ is the set of words that belong to the context of w , $c_{w,u}$ is a real number measuring the influence exercised by u over w , and $\lambda \in [0, 1]$ is a damping factor.

Eq. (1) defines the term weights by a sparse linear system of equations. In our experiments, the solution of such a system was computed by Jacobi algorithm, an efficient algorithm which can be applied even on huge problems with billions of variables [3].

Context Functions In order to define the influence factors, it has to be taken into account that words are multiply instantiated in different positions of a document, and each instance (a word-occurrence) is affected by a different context. Therefore, we distinguish between words, w , and word occurrences, \hat{w} . We assume that $c_{w,u}$ can be computed as the sum of the contributions of all the occurrences \hat{w} , \hat{u} of w and u , respectively, such that \hat{u} belongs to the context of \hat{w}

$$c_{w,u} = \sum_{\hat{w} \in occ(w)} \sum_{\hat{u} \in ict(\hat{w}, u)} C_p(\hat{w}, \hat{u}). \quad (2)$$

Here, $occ(w)$ is the set of instances of word w , and $ict(\hat{w}, u)$ is the set of the occurrences of u that belong to the context of \hat{w} , i.e. $ict(\hat{w}, u) = occ(u) \cap ctxt(\hat{w})$, where $ctxt(\hat{w})$ is the context of \hat{w} ; \mathbf{p} is a set of parameters, and $C_p(\hat{w}, \hat{u})$ is a parametric function that establishes the strength of the influence between the instances \hat{w} (the word under evaluation) and \hat{u} (the context word). In this work we define the context of a word $ctxt(\hat{w})$ as the set of words that are contained in the same document and within the surround of \hat{w} .

The function $C_p(\hat{w}, \hat{u})$ establishes how word couples can influence one another on the basis of features extracted from the words and from the relationships between words. This function can exploit any sort of feature from \hat{w} and \hat{u} : info-theoretical, morphological or lexical. The features that have been used in our preliminary experiments are exclusively statistical (Tab. 1).

The most general approach to the implementation of $C_p(\hat{w}, \hat{u})$ is by a modeling tool that has the universal approximation property, e.g. neural networks, polynomials, and rationales. For the introductory scope of this paper we preferred to adopt a simpler implementation of $C_p(\hat{w}, \hat{u})$. We defined the influence function as $C_p(\hat{w}, \hat{u}) = \prod_{i=0}^n \sigma(\alpha_i x_i + \beta_i)$, where x_i is the value associated with the i -th feature, α_i, β_i are the model parameters, and σ is the logistic sigmoid function $\sigma(x) = 1/(1 + e^{-x})$. Each term $\sigma(\alpha_i x_i + \beta_i)$ is a sort of soft switch related to the i -th feature and controlled by α_i (steepness and direction) and β_i (medium value) so that the whole function is a sort of boolean expression composed by AND operators.

¹ Dip. di Ingegneria dell'Informazione, Università di Siena, via Roma 56, 53100 - Siena - Italy, email: {ernandes, angelini, marco, rigutini, franco}@dii.unisi.it

| Name | Feature Set |
|-------|---|
| FS-A | $idf(w), idf(u), dist(\hat{w}, \hat{u})$ |
| FS-B | $idf(w), idf(u), dist(\hat{w}, \hat{u}), dist(\hat{u}, Q)$ |
| FS-B* | $idf(w), idf(u), dist(\hat{w}, \hat{u}), dist(\hat{u}, Q), sw-list$ |

Table 1. The set of features used by the Context Functions. $dist(\hat{w}, \hat{u})$ is the number of words between \hat{w} and \hat{u} , $dist(\hat{u}, Q)$ is the number words between \hat{u} and the word occurrences of query Q . Symbol *sw-list* denotes the cases that a stop word list is used to remove non informative terms.

Learning Context Functions The goal of the learning procedure is the optimization of a cost function e_p that measures the performance of the weighting system. We adopted the resilient parameter adaptation [4], which is an efficient technique that updates the parameters using the signs of the partial derivatives of the cost function.

The cost function e_p depends on the specific task that is addressed by the application. The mean reciprocal rank of the correct answer (MRR) is a standard positional and cumulative evaluation measure can be used for QA evaluation. Formally, given a set of queries $Q = \{q_1, \dots, q_n\}$ and denoted by $pos(a^q)$ the position of the correct answer a^q for question q , we have $MRR(Q) = \frac{1}{n} \sum_{q=1}^n \frac{1}{pos(a^q)}$.

On the other hand, MRR is a discrete function and cannot be used for gradient-based learning procedures. In order to design a differentiable cost function, we defined a ‘‘soft position’’ function $soft_pos$ to replaces pos : $soft_pos(a) = \sum_{w \in W, w \neq a} \sigma(\gamma(s(w) - s(a)))$, where W is the set of words considered by the weighting algorithm, a is the correct answer, γ is a predefined parameter and σ is a sigmoid function. In fact, it can be easily observed that for large γ , $soft_pos(a)$ approximates $pos(a)$. Finally, we defined a differentiable evaluation function which approximates MRR simply by replacing $pos(a)$ with $soft_pos(a)$. Once the system is trained using this soft function, the results can be measured by the standard MRR.

3 Experimental Results

The Single-Word QA problem The proposed method was experimentally assessed on Single-Word QA, a special case of QA where each question has to be answered with a single correct word, given a number of documents related to the question. A popular and challenging example of Single-Word QA is provided by crossword clues.

Our experiments aim to show that the Context-based algorithm improves the ranking quality of the candidate answers provided by WebCrow, a Web-based system designed to answer crossword clues [2].

The dataset consisted of 525 Italian crossword clues randomly extracted from the archive in [2]. The examples were divided into two subsets: *NE*-questions (165 examples), whose answers are factoid Named Entities, and, *nonNE*-questions (360 examples), whose answers are non-Named Entities (common nouns, adjectives, verbs, and so on. e.g. <Reduce the area by two: **foldinhalf**>).

Compared approaches The Context-based algorithm was compared with three different weighting techniques. The first is TFIDF, a standard measure that gives a statistical evaluation of the importance of a word to a document. The other two techniques, *WebCrow-S* and *WebCrow-SM*, are two ranking methods used in WebCrow. *WebCrow-S* exploits only statistical information of the words, the documents and the queries. *WebCrow-SM* additionally uses a morphological analysis of the documents joined with a morphological Answer Type classification of the clues (see [2]).

The experiments The subsets (*NE* and *nonNE*-questions) were randomly divided into a train set (40%) and test set (60%). Each example consisted of a question/answer couple, a set of H documents related by Google to the question; the vector of the ranked terms \mathbf{W} (the candidate answers), extracted from the H documents. Several

| Def TW | FS | NE | nonNE |
|----------------------------------|-------|--------------|--------------|
| TFIDF | | 0,216 | 0,071 |
| TFIDF + context reweighting | FS-A | 0,233 | - |
| WebCrow-S | | 0,290 | - |
| WebCrow-S + context reweighting | FS-B | 0,346 | - |
| WebCrow-S + context reweighting | FS-B* | 0,355 | - |
| WebCrow-SM | | 0,293 | 0,104 |
| WebCrow-SM + context reweighting | FS-B* | 0,345 | 0,121 |

Table 2. MRR performance. The results of the three baseline algorithms are given in bold. Def TW denotes the algorithm used for the default term weights d_w , FS specifies the Feature Set of the Context Function.

Context Functions with different set of features (Tab. 1) were trained. We chose 20 as a fixed dimension for the context window.

The achieved results, displayed in Table 2, confirm that the Context-based algorithm outperformed the other weighting techniques. For the *NE* subset (Table 2, *NE* column), the performance was impressive. In particular, the reweighting of the default scores provided by *WebCrow-S*, produced a 22,3% increment of the MRR (from 0,29 to 0,355, row 5). An insightful resume of the performance improvements is provided by the Success Rate curve (Fig. 1), that shows the probability of finding the correct answer within the first N words of \mathbf{W} . Each weighting scheme under comparison appears consistently below the curve of its Context-based reweighting.

4 Conclusions and Further work

In this paper we proposed a novel term weighting that exploits the information provided by word contexts. The approach has been proved to be very effective on the problem of Single-Word Web-based Question Answering. Future matter of research includes the application of the method to Document Classification and Information Extraction.

ACKNOWLEDGEMENTS

This work has been funded by the Google Research Awards Program. We are grateful to M. Bianchini, E. Di Iorio and G. Monfardini for their important advises and we thank M. Diligenti for his support.

REFERENCES

- [1] M. Berry, Z. Drmac, and E. Jessup, ‘Matrices, vector spaces, and information retrieval’, *SIAM Rev.*, **41**(2), 335–362, (1999).
- [2] M. Ermandes, G. Angelini, and M. Gori, ‘Webcrow: A web-based system for crossword solving.’, in *Proc. of AAAI ’05*, Pittsburgh, USA, (2005).
- [3] L. Page, S. Brin, R. Motwani, and T. Winograd, ‘The pagerank citation ranking: Bringing order to the web’, in *Proc. of ASIS ’98*, Pittsburgh, USA, (1998).
- [4] M. Riedmiller and H. Braun, ‘A direct adaptive method for faster back-propagation learning: The RPROP algorithm’, in *Proc. of ICNN ’93*, San Francisco, USA, (1993).
- [5] J. R. Seeley, ‘The net of reciprocal influence’, *Canadian Journal of Psychology*, **3**(4), 234–240, (1949).
- [6] E. Voorhees, ‘Natural language processing and information retrieval’, in *Proc. of SCIE ’99*, Frascati, Italy, (1999).

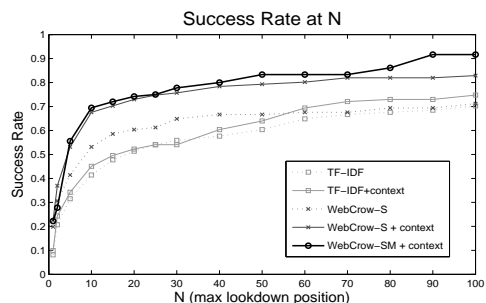


Figure 1. The probability (vertical axis) of finding the correct answer within the first N candidate answers (horizontal axis).