UniPadova @ LeQua 2022: A Preliminary Study of a BM25 Approach to Quantification

Giorgio Maria Di Nunzio^{1,2}

¹Department of Information Engineering, University of Padova, Italy ²Department of Mathematics, University of Padova, Italy

Abstract

Our participation to the LeQua lab continues the sequence of experiments dedicated to minimal coding that use the R Tidyverse packages to build reproducible source code for experiments in IR related tasks. In this specific case, we focused on the two-dimensional interpretation of the BM25 ranking formula that studies the distribution of documents on a two-dimensional space to study the quantification task without any type of optimization.

Keywords

Quantification, Two-dimensional BM25, R Tidyverse

1. Introduction

The Learning to Quantify lab (LeQua) at the Conference and Labs Evaluation Forum (CLEF) 2022, is the first edition of a laboratory dedicated to the evaluation of methods for text quantification [1]. The four subtasks available differ in the number of classes (binary or multi-class) and in the data source for training and testing (numerical matrices or raw textual documents).

Our participation to the LeQua lab continues the sequence of experiments dedicated to minimal coding that use the R Tidyverse¹ approach to build the software for the research setting and experimental analysis [2, 3]. In this specific case, our experiments focused on the two-dimensional interpretation of the BM25 ranking formula that studies the distribution of documents on a two-dimensional space to optimize the classification model [4].

Given the time constraints, we could participate only to Subtask T2A. This task focuses on the evaluation of binary quantifiers starting from raw text. For this reason, our main contribution with these experiments is summarized as follows:

- A study of a minimal binary quantifier based on BM25 without feature selection;
- A comparison with the QuaPy baseline w/out feature selection.

The remainder of the paper will introduce the methodology and a brief summary of the experimental settings that we used in order to create the official run submitted for this task.

CLEF 2022 – Conference and Labs of the Evaluation Forum, September 5–8, 2022, Bologna, Italy

[☆] giorgiomaria.dinunzio@unipd.it (G. M. Di Nunzio)

http://github.com/gmdn (G. M. Di Nunzio)

D 0000-0001-9709-6392 (G. M. Di Nunzio)

^{© 0 2022} Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

¹https://www.tidyverse.org

2. Method

Our method follows the principles described by [5] where a collection of textual document is processed through "pipelines", a sort of organized workflow split into steps where the output of one step is the input for the subsequent step. The R programming language has a set of packages named Tidyverse that implements this idea of pipelines. The additional value of this approach is that it is easily reproducible and easily readable.²

2.1. Pipeline for Importing Data

In order to preprocess the dataset of raw documents, we used the following pipeline:

- 1. start from raw data;
- 2. split text into words;³
- 3. transform to lowercase;
- 4. remove stopwords;
- 5. remove words with less than (or with exactly) two characters and keep only words without any number;
- 6. compute tf.

The corresponding six lines of code are shown in Listing 1 while the result of these passages for the ten most frequent words and their frequencies are shown in Listing 2.

Listing 1: Extraction of terms and term frequencies from raw data

```
1 raw_data %>%
```

```
2 unnest_tokens(word, text) %>%
```

- 3 mutate(word = tolower(word)) %>%
- 4 anti_join(get_stopwords()) %>%
- 5 filter (nchar (word) > 2 & str_detect (word, "^[a-z]+\$")) %>%
- 6 count(label, docid, word, name = "tf")

	Listing 2:	Top ten	term	and	their	frequencies
--	------------	---------	------	-----	-------	-------------

god	46
original	43
book	40
luke	36
artwork	29
new	28
acts	27
krsna	25
knife	25
maynard	24
	original book luke artwork new acts krsna knife

²The source code of the experiments is available at: https://github.com/gmdn. ³https://www.tidytextmining.com

2.2. Computing BM25

BM25 ranks documents according to the probability of relevance (R = 1) given a document d and a query q, P(R = 1|d, q). This probability can be approximated by the sum of the words w_i (see [6] for the derivation of the following equations):

$$P(R=1|d,q) \approx \sum_{t_i \in d \cap q} w_i \tag{1}$$

where

$$w_i = \frac{tf_i}{tf_i + K} \cdot \log\left(\frac{p_i}{q_i}\frac{(1-q_i)}{(1-p_i)}\right) , \qquad (2)$$

 p_i (or q_i) is the probability, estimated on the training data, that a relevant (or non-relevant) document contains the word w_i and tf_i is the term frequency of w_i in the document d and K a function of some parameters about the global statistics of the collection of documents.

In the two-dimensional representation of probabilities, we keep P(R = 1|d, q) distinct from the probability of a document being not relevant P(R = 0|d, q).⁴ For the quantification task, we will drop the variable q and turn this model into a classification problem where a document d is either member of a class or not (relevant or not for that class). In this way, the sum of the terms t_i (see Eq. (1)) is computed across all the terms of the document ($t_i \in d$).

With some algebraic manipulation — see [7] for an explanation of how these two parts can be derived from the original BM25 formulation — we obtain the following decision function (we hide some constant factors for a cleaner presentation):

$$\underbrace{\sum_{t_i} \log\left(\frac{p_i}{1-p_i}\right)}_{x} - \underbrace{\sum_{t_i} \log\left(\frac{q_i}{1-q_i}\right)}_{y} > 0 \tag{3}$$

If the inequality is true, the document is classified in the relevant category (R = 1), otherwise it w-ll be classified as non relevant (R = 0).

2.3. Plotting Predictions

The advantage of this BM25 reformulation is its geometric interpretation. The two sums, x and y, can be rendered as two coordinates in a two-dimensional space.⁵ In Figure 1, we show the result of the two-dimensional interpretation for the training set. Each point is a document and the two coordinates represent the BM25 weight decomposed in the two parts [8]. Points below the line are classified as 'positive' (class 1), points above the line as 'negative' (class 0). The color of each point shows the true label of the document. It is possible to see some misclassified

$$\log\left(\frac{P(R=1|d,q)}{P(R=0|d,q)}\right)$$

⁴While it is true that P(R = 1|d, q) = 1 - P(R = 0|d, q), there are probabilistic and implementation reasons that explain this more elaborate description [6]. In Eq. (1), the "approximetly equal to" derives from the fact that the documents ordered by the probability P(R = 1|d, q) are ordered in the same way by $\log \left(\frac{P(R=1|d,q)}{1-P(R=1|d,q)}\right) = \frac{P(R=1)}{1-P(R=1)}$

⁵In this paper, we are using x and y in traditional sense, i.e. coordinates of a two-dimensional Cartesian space, and not in a Machine Learning sense where x is the input and y is the output that the model has to predict.

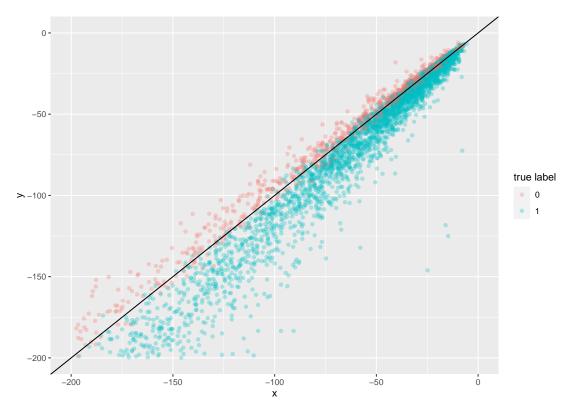


Figure 1: Visualization of the training documents on the two-dimensional space. Points below the line are classified as 'positive' (class 1), points above the line as 'negative' (class 0). The color of each point shows the true label of the document.

points along the line (a red dot below the line or a cyan dot above it). In quantification task, it is not important to have a perfect classifier (all the red points above the line and all the cyan points below it), but rather to keep these misclassified points balanced: the number of false positives should equal the number of false negatives to achieve a perfect the prediction in terms of proportions of the two classes.

3. Experiments

In our experiments, we run the two-dimensional BM25 and compare it with the QuaPy baseline trained with a Logistic Regression classifier and optimized with the all the classify and counts variants: CC, ACC, PCC, PACC.⁶

In the following items, we discuss our preliminary considerations about the analysis of the results:

• The unbalancedness in the training and development sets between the classes were never

⁶The Python code that we run ourselves to have the QuaPy baseline will be made available together with the R source code.

used to optof imize the probabilities the BM25.

- The number of unique terms we found is 41,905 which is almost four times the number of unique terms found with the QuaPy baseline (12,301 in total). If we filter the terms with document frequency greater or equal 3 (which is the default value for QuaPy) we get a comparable number of tokens (14,179).
- The absolute error AE on the training set (an overestimation of the goodness of the quantifier) is very small for BM25, AE = 0.0062. It is also interesting to note that the classifier is far from being perfect (accuracy of .85) but the proportion of false positives and false negatives is very good (348 against 379) without any optimization.
- The situation changes drastically in the development and in the test set, as shown in the overview results provided by the organizers of the Lab. We will investigate the amount of error due to the topic drift, change in the vocabulary, and the approach to count the proportions (in this version, we only performed a classify and count).
- We also found that the QuaPy tfidf baseline was different from the official results. For example, the best baseline with tfidf is PACC (we agree with this result)A the offEcial Relative absolute error is RAE = 0.138 (AE = 0.026) while our baseline is RAE = 0.294 (AE = 0.036), the worst tfidf baseline is PCC (while ours is CC), the official results are RAE = 1.362 (AE = 0.144) while our results are RAE = 2.615 (AE = 0.261).

4. Future Work

In the previous section, we highlighted some points that will drive our next steps. In particular, we know that the size of the vocabulary we used was much larger than the baselines, we also think that there is some additional feature selection that can give some additional value and level the results between the baselines. The additional part for improving the BM25 approach is the fact that we only used the classify and count (CC) approach which is almost the worst among all the baselines.

At the end, even though the results were not good at all, the exercise has been very stimulating for clarifying some steps, think about some new ideas of how to tackle the quantification approach, and see at work the QuaPy framework which is an excellent baseline for this task.

Acknowledgments

It is one of those cases where the anonymous reviewers were very engaging and made us think about how some parts of the paper could be improved and made clearer. We really want to thank them for all the comments and suggestions.

References

 A. Esuli, A. Moreo, F. Sebastiani, G. Sperduti, Overview of lequa 2022: Learning to quantify, in: Working Notes of the 2022 Conference and Labs of the Evaluation Forum (CLEF 2022), Bologna, IT, 2022.

- [2] G. M. D. Nunzio, A study on lemma vs stem for legal information retrieval using R tidyverse, in: P. Mehta, T. Mandl, P. Majumder, M. Mitra (Eds.), Working Notes of FIRE 2020 Forum for Information Retrieval Evaluation, Hyderabad, India, December 16-20, 2020, volume 2826 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2020, pp. 54–59. URL: http://ceur-ws.org/Vol-2826/T1-10.pdf.
- [3] G. M. D. Nunzio, As simple as possible: Using the R tidyverse for multilingual information extraction. IMS unipd ad CLEF ehealth 2020 task 1, in: L. Cappellato, C. Eickhoff, N. Ferro, A. Névéol (Eds.), Working Notes of CLEF 2020 - Conference and Labs of the Evaluation Forum, Thessaloniki, Greece, September 22-25, 2020, volume 2696 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2020. URL: http://ceur-ws.org/Vol-2696/paper_137.pdf.
- [4] G. M. D. Nunzio, Shiny on your crazy diagonal, in: R. Baeza-Yates, M. Lalmas, A. Moffat, B. A. Ribeiro-Neto (Eds.), Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, August 9-13, 2015, ACM, 2015, pp. 1031–1032. URL: https://doi.org/10.1145/2766462.2767867. doi:10.1145/ 2766462.2767867.
- [5] H. Wachsmuth, Text Analysis Pipelines Towards Ad-hoc Large-Scale Text Mining, volume 9383 of Lecture Notes in Computer Science, Springer, 2015. URL: https://doi.org/10.1007/ 978-3-319-25741-9. doi:10.1007/978-3-319-25741-9.
- [6] S. E. Robertson, H. Zaragoza, The probabilistic relevance framework: BM25 and beyond, Foundations and Trends in Information Retrieval 3 (2009) 333–389. URL: http://dx.doi.org/ 10.1561/1500000019. doi:10.1561/1500000019.
- [7] G. M. D. Nunzio, Geometric perspectives of the BM25, in: P. Boldi, R. Perego, F. Sebastiani (Eds.), Proceedings of the 6th Italian Information Retrieval Workshop, Cagliari, Italy, May 25-26, 2015, volume 1404 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2015. URL: http: //ceur-ws.org/Vol-1404/paper_4.pdf.
- [8] G. M. D. Nunzio, Interactive text categorisation: The geometry of likelihood spaces, in: C. Lai, A. Giuliani, G. Semeraro (Eds.), Information Filtering and Retrieval - DART@AI*IA 2014: Revised and Invited Papers, 8th International Workshop on Information Filtering and Retrieval, Pisa, Italy, December 10, 2014, volume 668 of *Studies in Computational Intelligence*, 2014, pp. 13–34. URL: https://doi.org/10.1007/978-3-319-46135-9_2. doi:10. 1007/978-3-319-46135-9_2.