Finding similar research papers using language models

Germán Hurtado Martín[1,2], Steven Schockaert[2], Chris Cornelis[2], and Helga Naessens[1]

[1] Dept. of Industrial Engineering, University College Ghent, Ghent, Belgium
[2] Dept. of Applied Mathematics and Computer Science, Ghent University, Ghent, Belgium

Abstract. The task of assessing the similarity of research papers is of interest in a variety of application contexts. It is a challenging task, however, as the full text of the papers is often not available, and similarity needs to be determined based on the papers' abstract, and some additional features such as authors, keywords, and journal. Our work explores the possibility of adapting language modeling techniques to this end. The basic strategy we pursue is to augment the information contained in the abstract by interpolating the corresponding language model with language models for the authors, keywords and journal of the paper. This strategy is then extended by finding topics and additionally interpolating with the resulting topic models. These topics are found using an adaptation of Latent Dirichlet Allocation (LDA), in which the keywords that were provided by the authors are used to guide the process.

Keywords: Similarity, Language modeling, Latent Dirichlet allocation

1 Introduction

Due to the rapidly growing number of published research results, searching for relevant papers can become a tedious task for researchers. In order to mitigate this problem, several solutions have been proposed, such as scientific article recommender systems [2, 8] or dedicated search engines such as Google Scholar. At the core of such systems lies the ability to measure to what extent two papers are similar, e.g. to find out whether a paper is similar to papers that are known to be of interest to the user, to explicitly allow users to find "Related articles" (as in Google Scholar), or to ensure that the list of search results that is presented to the user is sufficiently novel and diverse [3]. To find out whether two articles are similar, content-based approaches can be complemented with collaborative filtering techniques (e.g. based on CiteULike.org or Bibsonomy.org) or citation analysis (e.g. PageRank, HITS, etc.). While the latter are well-studied, content-based approaches are usually limited to baseline techniques such as using the cosine similarity between vector representations of the abstracts.

Comparing research papers is complicated by the fact that their full text is often not publicly available, and only the abstract along with some document

features such as keywords, authors, or journal can be accessed. The challenge thus becomes to make optimal use of this limited amount of information. Hurtado et al. [7] investigated the impact of individual document features within the vector space model. Their main conclusion was that baseline methods using only the abstract could not be improved significantly by enriching them with other features. Language modeling techniques, however, have been shown to perform well for comparing short text snippets [6, 10].

Our goal in this paper is therefore to explore how language models can be used to compare research paper abstracts, how they can best make use of the other document features, and whether they are a more reasonable choice than a vector space model based approach for this task. In particular, we combine two ideas to address these questions. On the one hand, we consider the idea of estimating language models for document features such as keywords, authors, and journal, and derive a language model for the article by interpolating them (an idea which has already proven useful for expert finding [11]). On the other hand, we apply LDA to discover latent topics in the documents, and explore how the keywords can help to improve the performance of standard LDA.

2 Research paper similarity

In this section we review and introduce several methods to measure article similarity, based on the information commonly available for a research paper: abstract, keywords, authors, and journal.

2.1 Vector space model

The similarity of two papers can easily be measured by comparing their abstracts in the vector space model (method *abstract* in the result tables): each paper is represented as a vector, in which each component corresponds to a term occurring in the collection. To calculate the weight for that term the standard tf-idf approach is used, after removing stopwords. The vectors $\mathbf{d_1}$ and $\mathbf{d_2}$ corresponding to different papers can then be compared using standard similarity measures such as the cosine (*cos*), generalized Jaccard (*g.jacc*), extended Jaccard(*e.jacc*), and Dice (*dice*) similarity, using them as in [7]. Alternatively, we also consider a vector representation where the abstract is completely ignored, and where there is one component for each keyword, with the weights calculated analogously as in the tf-idf model (method *keywords*).

In [7] an alternative scheme for using the keywords has been proposed, which does not ignore the information from the abstract. This scheme was referred to as explicit semantic analysis (ESA) since it is analogous to the approach from [4]. The idea is, departing from a vector $\mathbf{d}$ obtained by method *abstract*, to define a new vector representation $\mathbf{d_E}$ of this paper, with one component for every keyword k appearing in the collection. The weights of $\mathbf{d_E}$'s components are defined by $w_k = \mathbf{d} \cdot \mathbf{q_k}$, where the vector $\mathbf{q_k}$ is formed by the tf-idf weights

corresponding to the concatenation of the abstracts of all papers to which keyword k was assigned. This method is called *ESA-kws* in our experiments below. Similar methods are considered in which vector components refer to authors (*ESA-aut*) or to journals (*ESA-jou*). For efficiency and robustness, only authors are considered that appear in at least 4 papers in the *ESA-aut* method, and only keywords that appear in at least 6 papers in the *ESA-kw* method.

2.2 Language modeling

A different approach is to estimate unigram language models [9] for each document, and calculate their divergence. A document d is then assumed to be generated by a given model D. This model is estimated from the terms that occur in the abstract of d (and the rest of the abstracts in the collection). Using Jelinek-Mercer smoothing, the probability that model D generates term w is given by:

$$P^*(w|D) = \lambda P(w|d) + (1-\lambda)P(w|\mathcal{C}) \tag{1}$$

where $\mathcal{C}$ is the whole collection of abstracts. The probabilities $P(w|d)$ and $P(w|\mathcal{C})$ are estimated using maximum likelihood, e.g. $P(w|d)$ is the fraction of occurrences of term w in the abstract of document d. Once the models D_1 and D_2 corresponding to two documents d_1 and d_2 are estimated, we measure their difference using the well-known Kullback-Leibler divergence, defined by

$$KLD(D_1||D_2) = \sum_w D_1(w) log \frac{D_1(w)}{D_2(w)}$$

If a symmetric measure is desired, Jensen-Shannon divergence could alternatively be used.

Language model interpolation The probabilities in the model of a document are thus calculated using the abstracts in the collection. However, given the short length of the abstracts, we should make maximal use of all the available information, i.e. also consider the keywords k, authors a, and journal j. In particular, the idea of interpolating language models (also used for example in [11]), which underlies Jelinek-Mercer smoothing, can be generalized:

$$P^*(w|D) = \lambda_1 P(w|d) + \lambda_2 P(w|k) + \lambda_3 P(w|a) + \lambda_4 P(w|j) + \lambda_5 P(w|\mathcal{C}) \tag{2}$$

with $\sum_i \lambda_i = 1$. In order to estimate $P(w|k)$, $P(w|a)$, and $P(w|j)$, we consider an artificial document for each keyword k, author a and journal j corresponding to the concatenation the abstracts where k, a and j occur, respectively. Then, the probabilities are estimated using maximum likelihood, analogously to $P(w|d)$. Since a document may contain more than one author and one keyword, we define $P(w|k)$ and $P(w|a)$ as:

$$P(w|k) = \frac{1}{n}\sum_{i=1}^{n} P(w|k_i) \quad P(w|a) = \frac{1}{m}\sum_{j=1}^{m} P(w|a_j) \tag{3}$$

where n and m are the number of keywords and authors in the document.

Latent Dirichlet Allocation Two conceptually related abstracts may contain different terms (e.g. synonyms, misspellings, related terms), and may therefore not be recognized as similar. While this is a typical problem in information retrieval, it is aggravated here due to the short length of abstracts. To cope with this, methods can be used that recognize which topics are covered by an abstract, ignoring the actual terms that occur. The idea is that topics are broader than keywords, but still sufficiently discriminative to yield a meaningful description of the content of an abstract. This topical information is not directly available; however, it can be estimated by using Latent Dirichlet Allocation (LDA) [1].

The idea behind LDA is that documents are generated by a (latent) set of topics, which are modeled as a probability distribution over terms. To generate a document, a distribution over those topics is set, and then, to generate each word w in the document, a topic z is sampled from the topic distribution, and w is sampled from the word distribution of the selected topic. In other words, the set of distributions ϕ over the words in the collection and the set of distributions θ over all the topics must be estimated. To do so, we use LDA with Gibbs sampling [5]. We can then estimate these probabilities as:

$$P(w|z) = \hat{\phi}_z^{(w)} = \frac{{n_z}^{(w)} + \beta}{{n_z}^{(\cdot)} + W\beta} \tag{4}$$

$$P(z|t) = \hat{\theta}_z^{(d)} = \frac{{n_z}^{(d)} + \alpha}{{n_\cdot}^{(d)} + T\alpha} \tag{5}$$

where t is the LDA model obtained with Gibbs sampling, W is the number of words in the collection, and T is the number of topics. Parameters α and β intuitively specify how close (4) and (5) are to a maximum likelihood estimation. The count ${n_z}^{(w)}$ is the number of times word w has been assigned to topic z, while ${n_z}^{(d)}$ is the number of times a word of document d has been assigned to topic z. Finally, ${n_z}^{(\cdot)}$ is the total number of words assigned to topic z, and ${n_\cdot}^{(d)}$ is the total number of words of document d assigned to any topic. All these values are unknown a priori; however, by using Gibbs sampling they can be estimated.

To find the underlying topics, the LDA algorithm needs some input, namely the number T of topics to be found. Based on preliminary results, we set $T = K/10$, where K is the number of keywords that are considered. The topics that are obtained from LDA can be used to improve the language model of a given document d. In particular, we propose to add $P(w|t)$ to the right-hand side of (2), with the appropriate weight λ. $P(w|t)$ reflects the probability that term w is generated by the topics underlying document d. It can be estimated by considering that:

$$P(w|t) = \sum_{i=1}^{T} P(w|z_i) \times P(z_i|t) \tag{6}$$

This method is referred to as *LM0* in the result tables.

The method *LM0* can be improved by taking advantage of the keywords that have been assigned to each paper. In particular, we propose to initialize the topics by determining T clusters of keywords using k-means. Then, a document

c is created for every cluster. This artificial document is the concatenation of the abstracts of all papers to which some keyword from the cluster was assigned. Once these documents c are made, initial values for the parameters ${n_z}^{(w)}$, ${n_z}^{(d)}$, and ${n_z}^{(\cdot)}$ in (4) and (5) can be retrieved from them: ${n_z}^{(w)}$ is initialized with the number of occurrences of w in artificial document c_z, ${n_z}^{(d)}$ with the number of words of document d occurring in c_z, and ${n_z}^{(\cdot)}$ with the total number of words in c_z. Parameter $n_{\cdot}^{(d)}$ is independent from the clustering results as it takes the value of the total number of words in document d. We furthermore take $\alpha = 50/T$ and $\beta = 0.1$. Subsequently, we can either work directly with these initial values (i.e., use them in (4) and (5): method *LM1* in the results tables), or we can apply Gibbs sampling (method *LM2*). In this last case, the values resulting from the clustering are only used to initialize the sampler, as an alternative to the random values used normally.

3 Experimental evaluation

3.1 Experimental Set-Up

To build a test collection and evaluate the proposed methods, we downloaded a portion of the ISI Web of Science[3], consisting of files with information about articles from 19 journals in the Artificial Intelligence domain. These files contain, among other data, the abstract, authors, journal, and keywords freely chosen by the authors. A total of 25964 paper descriptions were retrieved, although our experiments are restricted to the 16597 papers for which none of the considered fields is empty.

The ground truth for our experiments is based on annotations made by 3 experts. First, 100 articles with which at least one of the experts was sufficiently familiar were selected. Then, using tf-idf with cosine similarity, the 30 most similar articles in the test collection were found for each of the 100 articles. Each of those 30 articles were manually tagged by the expert as similar or dissimilar. To evaluate the performance of the methods, each paper $\mathbf{p}$ is thus compared against 30 others[4], some of which are tagged as similar. Similarity measures can then be used to rank the 30 papers, such that ideally the papers similar to $\mathbf{p}$ appear at the top of the ranking. In principle, we thus obtain 100 rankings. However, due to the fact that some of the lists contained only dissimilar articles, and that sometimes the experts were not certain about the similarity of some items, the initial 100-article set was reduced to 89 rankings. To evaluate these rankings, we use mean average precision (MAP) and mean reciprocal rank (MRR).

Table 1. Vector space model

MAP				
	cos	*dice*	*e.jacc*	*g.jacc*
abstract	0.492	0.492	0.492	0.543
keywords	0.496	0.497	0.495	0.481
ESA-kws	0.544	0.544	0.543	0.537
ESA-aut	0.553	0.553	0.553	0.554
ESA-jou	0.414	0.414	0.414	0.422

MRR				
	cos	*dice*	*e.jacc*	*g.jacc*
abstract	0.689	0.689	0.689	0.725
keywords	0.736	0.741	0.738	0.715
ESA-kws	0.701	0.701	0.701	0.717
ESA-aut	0.726	0.726	0.726	0.727
ESA-jou	0.574	0.574	0.574	0.581

Table 2. Language modeling

	MAP			MRR		
λ-config.	LM0	LM1	LM2	LM0	LM1	LM2
0.9 0 0 0 0	0.596	0.596	0.596	0.759	0.759	0.759
0 0.9 0 0 0	0.533	0.533	0.533	0.732	0.732	0.732
0 0 0.9 0 0	0.520	0.520	0.520	0.664	0.664	0.664
0 0 0 0.9 0	0.294	0.294	0.294	0.395	0.395	0.395
0 0 0 0 0.9	0.383	0.365	0.522	0.547	0.500	0.693
0.7 0 0 0 0.2	0.607	0.602	0.622	0.765	0.771	0.766
0.5 0 0 0 0.4	0.602	0.604	0.636	0.745	0.769	0.778
0.2 0 0 0 0.7	0.585	0.607	0.613	0.746	0.764	0.759
0.7 0.2 0 0 0	0.593	0.593	0.593	0.786	0.786	0.786
0.5 0.4 0 0 0	0.575	0.575	0.575	0.781	0.781	0.781
0.2 0.7 0 0 0	0.544	0.544	0.544	0.736	0.736	0.736
0.4 0.1 0 0 0.4	0.632	0.650	**0.671**	0.803	**0.821**	0.820
0.1 0.4 0 0 0.4	0.558	0.560	0.593	0.748	0.754	0.775
0.4 0.4 0 0 0.1	0.576	0.578	0.588	0.773	0.775	0.782
0.3 0.3 0 0 0.3	0.584	0.591	0.618	0.772	0.788	0.797
0.3 0.1 0.1 0.1 0.3	0.614	0.617	0.654	0.778	0.780	0.784
0.4 0.1 0.1 0 0.3	0.615	0.620	0.656	0.770	0.779	0.800
0.4 0.1 0 0.1 0.3	0.637	0.644	0.667	0.811	0.814	0.824

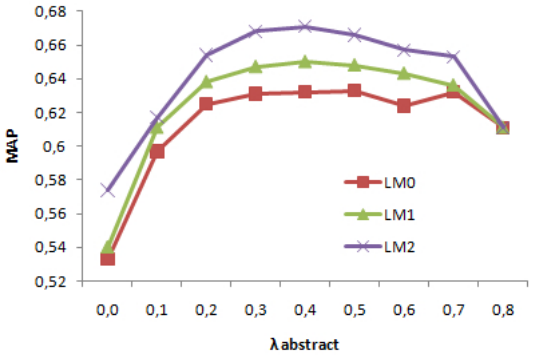

Fig. 1. Importance of abstract vs. topics

3.2 Results

Tables 1 and 2 summarize the results of the experiment. The λ configurations in the first column of Table 2 are shown in the order $\lambda_{abstract}$, $\lambda_{keywords}$, $\lambda_{authors}$, $\lambda_{journal}$, λ_{topics}. We fixed the sum of these weights to 0.9, and set the general smoothing factor (λ_5 in (2)) to 0.1.

The main conclusion that we can draw from these results is that language models are indeed capable of yielding a substantial improvement over all of the vector space approaches. The first block of Table 2 summarizes the results obtained with language models that only use one of the features. We find that language models which only use the abstract significantly[5] improve the performance of the most traditional vector space methods (*abstract*). Models uniquely based on other features can perform slightly better than *abstract*, but these improvements were not found to be significant. However, these results are still useful as an indication of the amount of information contained in each of the features: language models based exclusively on keywords or on authors perform comparable to the method *abstract*. Using topics only yields such results when *LM2* is used, while the information contained in the journal is clearly poorer.

In the second block of Table 2 we examine different combinations of two features: abstract with topics on the first three lines, and abstract with keywords on the last three. These results confirm that the abstract contains the most information, and should be assigned a high weight. On the other hand, we can observe how the topics, when combined with the abstract, yield a better MAP score. In particular, the MAP score for the LM2 configuration on the second

[3] http://apps.isiknowledge.com

[4] During the annotation process it was also possible to tag some items as "Don't know" for those cases where the expert had no certainty about the similarity. These items are ignored and therefore some papers are compared to less than 30 others.

[5] In this work we consider an improvement to be significant when $p < 0.05$ for the paired t-test.

(resp. third) line of the second block are significantly better than the LM2 score on the fifth (resp. sixth) line.

The third block of Table 2 shows the results of combining abstract and topics, with keywords, authors, and journal. It is clear that giving a small weight to keywords is beneficial, as it leads to the highest scores, which are significantly better than all configurations of the second block. For authors and journal, however, we do not find a substantial improvement. In Fig. 1 we further explore the importance of the abstract and the topics. We set the weight of the keywords to a fixed value of 0.1, and the remaining weight of 0.8 is divided between abstract and topics. What is particularly noticeable is that ignoring the abstract is penalized stronger than ignoring the topics, but the optimal performance is obtained when both features are given approximately the same weight.

Finally, we can note from Table 2 that *LM1* only slightly improves *LM0*, but larger differences in MAP scores can be observed between *LM1* and *LM2*. These latter differences are significant for all configurations in the third block, and the two first configurations in the second block. The difference is particularly striking when only the topics are used to create the models ($\lambda_{topics} = 0.9$), which shows how much LDA can benefit from the initialization based on the keywords.

4 Conclusion

We have shown how language models can be used to compare research paper abstracts and how their performance for this task can be improved by using other available document features such as keywords, authors, and journal. In particular, language models have proven more suitable in this context than any of the vector space methods we considered. We have also explored how LDA could be used in this case to discover latent topics, and a method has been proposed to effectively exploit the keywords to significantly improve the performance of the standard LDA algorithm.

References

1. D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.
2. T. Bogers and A. van den Bosch. Recommending scientific articles using CiteULike. In *Proceedings of the 2008 ACM conference on Recommender systems*, pages 287–290, 2008.
3. C. L. Clarke, M. Kolla, G. V. Cormack, O. Vechtomova, A. Ashkan, S. Büttcher, and I. MacKinnon. Novelty and diversity in information retrieval evaluation. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 659–666, 2008.
4. E. Gabrilovich and S. Markovitch. Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, pages 1606–1611, 2007.
5. T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(Suppl. 1):5228–5235, 2004.

6. L. Hong and B. D. Davison. Empirical study of topic modeling in Twitter. In *Proceedings of the First Workshop on Social Media Analytics*, pages 80–88, 2010.
7. G. Hurtado Martín, S. Schockaert, C. Cornelis, and H. Naessens. Metadata impact on research paper similarity. In *Proceedings of the 14th European Conference on Research and Advanced Technology for Digital Libraries*, pages 457–460, 2010.
8. S. M. McNee, I. Albert, D. Cosley, P. Gopalkrishnan, S. K. Lam, A. M. Rashid, J. A. Konstan, and J. Riedl. On the recommending of citations for research papers. In *Proceedings of the 2002 ACM Conference on Computer Supported Cooperative Work*, pages 116–125, 2002.
9. J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 275–281, 1998.
10. X. Quan, G. Liu, Z. Lu, X. Ni, and L. Wenyin. Short text similarity based on probabilistic topics. *Knowledge and Information Systems*, 25:473–491, 2010.
11. J. Zhu, X. Huang, D. Song, and S. Rüger. Integrating multiple document features in language models for expert finding. *Knowledge and Information Systems*, 23:29–54, 2010.