DMINR at TREC News Track

Sondess Missaoui, Andrew MacFarlane

Stephann Makri, Marisela Gutierrez-Lopez

Centre for HCI Design
City, University of London, London, England, UK
sondess.missaoui@city.ac.uk, A.Macfarlane-1@city.ac.uk,
Stephann.Makri.1@city.ac.uk, Marisela.Gutierrez-Lopez@city.ac.uk

Abstract. This paper describes the use of the DMINR Named entity extraction and linking system in TREC 2019. The track entered for are: News track, involves both Background Linking and Entity Ranking. Our approach to each of these tasks draws on prior work done by City, University of London at the TREC conference. In the background linking task, we treated the problems as an adhoc search task, using Named Entities (NEs) from a set of documents identified in pseudo relevance feedback, and optimizing these using a Hill-Climbing algorithm to provide a set of related articles. In the Entity Ranking task, we compared an approach using the BM25 ranking method with a probabilistic model that uses Wikipedia data as a resource to rank entities. The probabilistic model utilises an entity modeling approach to disambiguate NEs. Then , it utilises a scoring model which uses the given entities to provide a score for them based on evidence from the news articles.

1 Introduction

This paper describes our participation in the Background Linking and Entity Ranking tasks of the TREC 2019 News Track- the second competitive evaluation campaign for IR systems in adversarial online news domain. This track focused on two complementary tasks (i.e., Background Linking and Entity Ranking) that aim to help the user understand the context of a news story, wherever they are reading it. We focus on the users' understanding of a story in relation to key entities for that story.

The participants were asked to (1) recommend articles that provided context and background information for the given topic (i.e., query article) and (2) determine which entities are linkable for the given article and rank them in terms of "usefulness", i.e., according to their influence on user's understanding of the topic. The TREC Washington Post Collection, five years of articles (2012 -2017), were provided for the development and evaluation of systems that participated in this benchmarking activity.

In this year's News Track, we focus on both tasks and propose methods that use *'focused' Named Entities* (NEs) as leads to find the background articles; and to rank provided entities.

More specifically, our proposed methods are based on the structure of news articles. Given, our experience in working in the journalism domain within DMINR project[1], we assert that a news article usually has a main story, often reported as an event that can be effectively summarised by the five *Ws*: *Who, What, When, Where and Why*. Many of the five Ws can be associated with appropriate Named Entities in the article. Moreover, an article encompasses different aspects that explain how the event happened. These aspects are represented through a set of concepts, artifacts, and entities. All this information is important in providing a background to the news story as well as for ranking NEs based on their "usefulness". In all the proposed methods, we focused in using most *topical named entities* among all entities in the query article and/or recommended articles. These focused named entities have proven to be useful for many natural language processing applications, including summarization, topic identification, and background linking [18].

In the next section, we will explain our assumptions and our methods for identifying and using focused NEs for both News Track tasks.

[1] A *Google News Initiative* (DNI) funded R&D project(see https://blogs.city.ac.uk/dminr)

2 Prior work at City: OKAPI and PLIERS at TREC

City has long been associated with the TREC conference. In our work on the News Track, the prior work at City was reviewed in order to find appropriate methods to use in the News track tasks, and it was clear that there were some ideas that were highly useful in this context. This relates both to the OKAPI work [1, 15, 16] and MacFarlane's PhD work done with the PLIERS system [9, 10]. This work developed a series of hill-climbing algorithms (simple heuristic learning methods) to address the selective dissemination of information problem dealt with in the TREC routing [6] and filtering tracks [7]. These methods take the basic assumption that relevance assessments are available for a given topic, and that these relevant documents can be used to extract a set of useful terms to represent the users' profiles e.g. using the *Robertson term selection value* (RSV) to rank terms based on probabilistic techniques [14]. This simple technique can be enhanced by optimizing on the term set. The top 300 terms or so can be chosen, but this is very costly. There is, however, clear evidence that optimizing on the topic 20-30 terms is sufficient [11].

2.1 Hill Climbing algorithms

Once a set of terms has been chosen by the initial relevance feedback mechanism (or pseudo relevant feedback method), the terms set can be optimized in various ways; it is a good idea to start with a set of seed terms for the algorithm i.e. the top 3 ranked terms from in initial term selection stage. There are many ways of doing this using various machine learning techniques e.g. deep learning [13], but there is evidence that Hill Climbers are much quicker than other schemes and provide the same level of retrieval effectiveness. MacFarlane et al. [11] compared Genetic Algorithms with Hill Climbers demonstrating this clearly. There are four algorithms that can be considered as follows:

1. *Find Best* (FB): An iteration examines the whole term set available and chooses the best term from that set. Iterations are long and the method can be computationally expensive.
2. *Choose First Positive* (CFP): An iteration stops when a single term shows sufficient improvement on the set already chosen. In this method the number of iterations is increased, but each is quite quick in comparison.
3. *Choose All Positive* (CAP): All terms that improve the term set within one iteration are accumulated and kept in the terms set for the next iteration. In this method the iterations are much longer, but there are many few of them.
4. *Choose Some Positive* (CSP): This is a compromise between CFP and CAP. A maximum number of terms per iteration can be selected (e.g. 3) and the iteration stops and moves to the next iteration. Computationally this algorithm is a half-way house.

The most suitable choice depends on the context e.g. document collection, topics etc. There does not appear to be a single best algorithm: in OKAPI at TREC-3 CAP produced the best results [16]. However, FB was found to be the most useful with PLIERS at TREC-8 [10]. In each of these algorithms, different term selection strategies can be used within a given iteration: terms can be added only, terms can be deleted only -or terms can be added or removed. In practical terms, adding and/or removing terms is most useful as it allows more term combinations to be examined. Further schemes on weights can also be considered in conjunction with the term selection strategy (see section 2.2 below). Because the algorithms are heuristics, some form of stopping criteria needs to be set otherwise the algorithms may keep going forever (or until the computer crashes). Several criteria can be used including:

1. A maximum number of iterations for positive term additions (e.g. 3 iterations where results of adding terms were no better or worse).
2. A threshold for improvements on term addition has been reached (e.g. an improvement in evaluation score below 0.01).
3. The elapsed time for term selection as exceeded a given maximum length (e.g. 2 hours).
4. A maximum number of terms is set (e.g. 20 terms).

Any or all of these stopping criteria can be used.

2.2 Term weighting schemes

In all the prior work at City, the underlying scheme used for term weighting was the BM25 model [1, 9, 10, 15, 16]. It is entirely possible that machine learning heuristics could be applied to learning the weightings for each term e.g. learning to rank [2], but this would be even more time consuming. In practice the City work either used the BM25 weight or attempted to reduced or increase the term by a third when adding a given term to the term set. This simple scheme was more than sufficient for requirements.

2.3 Metrics for evaluation and optimization

A score needs to be generated in order to evaluate the improvement or otherwise of applying that term (adding, removing, and reweighting). A number of metrics were used at OKAPI at TREC 4 [15] -including average precision (AVEP), which proved to be the best predictor of results; AVEP improved the results on other metrics, but other metrics did not provide the same level of prediction. This work was carried about before the B-PREF and nDCG metrics were advocated, but such metrics could also be used for term selection optimization using any of the term selection algorithms outlined in section 2.1 above.

3 Query Expansion for Background Linking: Using Named Entities as leads

In this section, we describe our approach to the Background Linking task in news domain. We developed a baseline system that does not rely on any external sources of evidence.

3.1 Methodology

Our method treats the Background Linking task as an adhoc search task, for which *Named Entities* (NEs) are extracted from the query article, and these are then used as leads to find background articles, i.e., retrieve and recommend relevant news stories.

As discussed above (see section 1), we assume that NEs within the query article represent core aspects of the news story. Therefore, the relatedness between a named entity and a background article is similar to the relevance concept in Information Retrieval (IR). In this method, recommending background articles is tackled as a retrieval problem. We extracted a set of useful NEs from the relevant documents and optimized on them for query expansion based ranking method [17]. In particular, we used pseudo-relevance feedback for each query article, i.e., topic, to extract a set of relevant NEs. Then, we used the NEs to expand the search query and retrieve relevant articles as a background, covering a diverse set of aspects of the original story.

As news articles may encompass many NEs, expanding the search query using all NEs will make the query too long. Therefore using it directly for retrieval might lead to noisy results [4, 8]. Given that, distinguishing *focused NEs*, topical NEs that are most representative of the meaning of the news article, is fundamental to expand the initial query and retrieve relevant results. A key question for us is therefore: how to identify the *focused NEs* from within the pseudo-relevance feedback?

We argue that this problem is a term selection optimization problem for query expansion in information retrieval and we tackle it using a heuristic learning method, i.e., hill-climbers [11]. This is a two-stage process: (i) apply the Robertson term selection value to rank NEs and extract a set of candidate named entities to represent the news story; and (ii) optimize on the NEs set using a "Find Best" extension of Hill Climber algorithm (see section 2).

In the first stage, we select a set of candidate named entities, that we define as "*focused NEs*", the NEs which maximize Robertson's Term Selection Value [14] using the following term selection scheme:

$$TSV_{e_i} = R * W_{rsi} \tag{1}$$

Where R is the number of pseudo-relevant documents, W_{rsi} is the Robertson's weight given by:

$$W_{rsi} = log\left[((r + 0.5) * (N - n - R + r + 0.5)) \,/\, ((n - r + 0.5) * (R - r + 0.5))\right] \tag{2}$$

where r is the number of pseudo-relevant documents where the entity e_i occurs in; R is the number of pseudo-relevant document for the query article. n is the number of documents entity e_i occurs in; N is the number of documents in the collection.

In the second stage, we ran the Hill climbers, on the set of top k entities selected based on TSV_{e_i} [11]. The fitness function used for Hill climber optimization was the standard TREC average precision measure (AVEP).

3.2 Submitted Run

Pre-processing of the datasets All of our methods largely rely on the Elasticsearch search engine [12] for indexing the dataset, i.e., the Washington Post collection provided by TREC.

In method, $cityuni\_1$, we used the classic tokenizer, a grammar based tokenizer, for which words were lowercased but not stemmed. We treated all fields from the JSON documents separately. For the retrieval model, we relied on the default scoring method in Elasticsearch, which is BM25.

For Named Entities Recognition, there are many existing approaches including machine learning techniques e.g. Stanford NER [3], but SpaCy demonstrated good performance in term of accuracy and run time in the datasets. We employed the SpaCy[2] tool to automatically annotate the news story's text (including the title and all the contents fields).

Background Linking This method makes the basic assumption that relevance assessments are available for the given topic. However, this is not the case for 2019 Topics. To overcome this limitation, we followed a query formulation scheme using the topic's title. For each topic we employ the NEs extracted from the title as query Q, then, we retrieve a set of relevant documents that are used as relevance feedback. The top-m documents, with m = 10, provide a pool from which candidate query terms , i.e., in our case the *query NEs*, are chosen based on the term selection value following eq.1. Based on our previous research [11], we determined that optimizing on a set of 30 NEs is sufficient for the value of K. Then, based on evidence from our previous work [10], we applied the "Find Best" algorithm to choose the best NEs from the k elements. The algorithm started with the top three ranked NEs for each topic to calculate the mean average precision measure; Then, it selected a NE if it yielded an increase in mean average precision. For instance, we adopted a selection strategy where NEs are only added. Moreover, a maximum number of NEs equal to 20 was set as the stopping criteria.

Finally, NEs that were selected from relevance feedback, were employed to expand a query Q. A new query Q_e is then formulated and used to retrieve background articles using Elasticsearch as the retrieval system. For the length of expanded query Q_e, we set $|Q_e|$ = 25 (without considering the rank). After retrieving relevant documents, the final step is to filter the results, by removing all the articles from the "Opinion", "Letters to the Editor", or "The Post's View" categories, as labeled in the "kicker" field.

4 Entity Ranking Task

Given a topic with title and content that is annotated with a list of mentioned entities in the article, i.e., a set of entities $E = e_1, e_2, ...e_n$, the task is to rank the predefined entities by importance for the news article. In this section, we describe the multiple approaches taken by the DMINR team towards the entity ranking task. Run 1, $City\_ER1$, is anchored in a BM25 based ranking methodology. While run 2, $City\_ER2$, is based on a probabilistic model that uses the Wikipedia Dump as external resources to rank the entities.

4.1 BM25 for Entities Ranking

In this method, we rank the set of predefined Named Entities based on how often they appear in each News story compared to all other NEs in the document, using the BM25 model. In particular, we rank the mentioned Named Entities, based on how much they are representative of the meaning of the News Article. We define these NEs as the key NEs that maximize the BM25 score.

[2] https://spacy.io/

As with the Background Linking task, no external sources of evidence were deployed. We followed the below steps:

- Pre-processing and indexing the datasets using Elasticsearch (i.e., same settings as the Background Linking task).
- Analyzing the input News story by performing a simple Named Entities Recognition algorithm. More precisely, extracting NEs using SpaCy.
- Ranking the list of mentioned Named Entities against the documents (represented as a set of NEs) using BM25 score.

4.2 Wikipedia as a Knowledge Base for Entities Ranking

Our second method, operating on the index built using Elasticsearch, relies on the Wikipedia dump to rank the Named Entities. In particular, the Wikipedia dump is used as the *Knowledge Base* (KB) [5] to represent the list of mentioned entities.

Methodology Our approach assumed that the features to be used to rank entities also require knowledge on the entities semantic meaning and how this meaning is representative of the news story. We also assumed that entities are produced from a mixture of mention strings. Those mentions are highly relevant for matching an entity against a news article. For this reason, we developed a twofold-based approach: (i) an 'entity modeling' method (see algo.1) to map the entities to their Wikipedia Dump. (ii) a 'scoring model' that uses the entity model to score it against the news article text (more details are provided below).

The 'entity model' is a mapping model that grounds entity mentions to their corresponding node in a Knowledge Base (i.e., Wikipedia Dump in this experiment), which allows the representation of an entity e through a set of terms/entities. The main motivation behind the 'entity modeling' method, is that entities can be referred to by many mention strings inside the news article, and the same mention string may be used to refer to multiple entities. Hence, the mapping method will identify information about entities that help to disambiguate them and spot them semantically within the news article.

The second step, the 'scoring model' represents a topic as a mixture over an underlying set of *Named Entities* probabilities using a Bayesian model. Furthermore, the NEs probabilities provide an explicit representation of the News article, which allows ranking according to their importance. In this approach, all our model's components are represented as a set of *Named Entities*. Formally, we define:

- Wikipedia Dump denoted as $Wiki$ is used as a Knowledge Base to enhance the entity semantic representation.
- A *Named Entity* is a sequence of n "similar" entities (i.e., semantically similar) extracted from its corresponding node in the *Knowledge Base* (i.e., Wikipedia Dump) and denoted by e, $e \leftarrow S_e$ where $S_e = s_1, s_2, ..., s_n$ (i.e., list of similar entities extracted from $WiKi$), where s_n is the nth entity in the sequence.
- News article is a document, a sequence of m *Named Entities* denoted by $D = e_{w_1}, e_{w_2}, ..., e_{w_m}$

Using these defined elements, we score a Named Entity e based on its probability to fit in the News article D, when it occurs in the Wikipedia Dump $Wiki$. Formally:

$$score(e, D) = P\left(D|e, Wiki\right) = \frac{tf\left(e, D\right) * tf\left(Wiki, D\right)}{|D|}$$

(3)

Where $tf\left(e, D\right)$ denotes the frequency of e in D. Given, $e \leftarrow S_e$, this frequency is cumulative for all $s_i \in S_e$. Formally:

$$tf\left(e, D\right) = \sum_{i=1}^{n} tf\left(s_i, D\right) \quad (4)$$

Similarly, $tf\,(Wiki, D)$ in eq.3 denotes the co-occurence between D and $wiki$, it is the count of *Named Entities* in $Wiki$ that belongs to D, and $|D|$ is total count of *Named Entities* in D

$$tf\,(Wiki, D) = \sum_{j=1}^{p} tf\,(t_j)\,, t_j \in \{D \cap Wiki\} \tag{5}$$

Submitted Run Run 2, $City\_ER2$, of which steps are summarized in Algorithms 1 and 2, shares the same initial steps with $City\_ER1$, i.e., the dataset pre-processing and named entities recognition scheme is identical. Each topic in the Entity Ranking task consists of a news article D from the Washington post, that is linked to a mentioned list of entities E. Moreover, each entity $e \in E$ is linked to its Wikipedia identifier (as explained above) and modeled as described in the following algorithm 1.

```
Algorithm 1: Entity Modeling Algorithm
  Result: Entity model
  Input: Named Entity mention ($e$); Wikipedia Dump ($W$)
  procedure MODEL($e, W$)
      index $W_e$                                    ▷ The $W_e$ is the Wiki. dump provided for $e$
      extract the set of Named Entities $S_{W_e}$ from $W_e$
      Rank $S_{W_e}$ using the RSV (see eq. 1)
      extract $S_e$ set of top 20 NEs in $S_{W_e}$
      $e \leftarrow S_e$
      return $S_e$
  end procedure
```

After the 'entity modelling' step is realised, we exploit the entities models in the Entities Ranking task as described in algorithm 2

```
Algorithm 2: $City\_ER2$, Named Entities Ranking Algorithm
  Result: ranked list of NEs
  initialization;
  while $e \in E$ do
      model $e$ (Algo.:1);
      Calculate relevance score for $e$ given $D$ (Eq.: 3);
      if $score \geq \delta\ threshold$ then
          add $E_{final}$ ←e
      end
      Rank $E_{final}$
  end
```

5 Evaluation Results

5.1 Overall results

At the time of writing, runs for other TREC News Track participants are not yet available. Hence, we just report results of our runs in the Entity Ranking and Background Linking tasks without any baseline comparison. The effectiveness of the submitted runs are reported in Tables 1 and 2. We focus on nDCG@5 and P@5 as a primary effectiveness measures across all topics. For the Background Linking task, our method achieves a relatively low nDCG@5 score. This can be explained by the fact it relies mainly on pseudo-relevance feedback as relevance judgements are not not available for this year topics. To address this problem, we used NEs extracted from the topic's title to retrieve a pool for pseudo-relevance feedback. However, the title contains only few named entities, yielding shorter queries, losing the gist of the input topic. This arguably sacrifices some retrieval effectiveness.

We noticed that our second method City_ER2 for the Entity Ranking task , which relies on the Wikipedia Dump outperforms the first method City_ER1. This can be explained by the fact that the KB-based representation of NEs allows to semantically evaluate the NEs relevance within the news article.

Table 1: nDCG@5 and P@5 for submitted runs for Entity Ranking task

Run	nDCG@5	P@5
City_ER1	0.5158	0.4115
City_ER2	0.5582	0.4538
median	0.537	0.4326

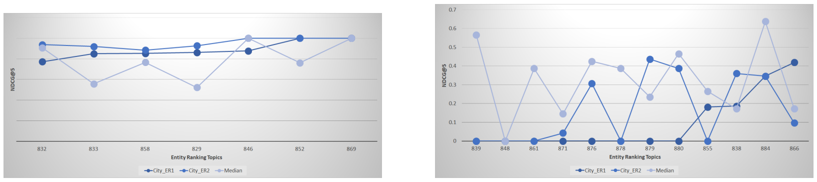
Table 2: nDCG and Precision for the Background Linking Runs

Run	nDCG@5	P@5
city_uni1	0.3251	0.4982

5.2 Failure Analysis

The overall results of our methods in both tasks are close to the median performance of all submitted runs to the track in terms of nDCG@5. The main observation is that our hypothesis that using entity as a main feature for both tasks (Entity Ranking and Background Linking) seems to be supported. However, our runs perform poorly for some topics. To clarify in which cases our methods work the best, we performed a failure analysis. In particular, for each of our runs, we identify the topic for which the related method achieves the highest gain or loss in terms of nDCG@5 comparing to the median.

For the Entity Ranking task, our methods $City\_ER1$ and $City\_ER2$ achieve their highest nDCG@5 for over the same 7 topics as Shown in Fig 1(a), where the biggest gain was in topic '*852*' with 31.2% over the median (0.76215). In the other hand, both methods achieve the highest loss comparing to the median with two different topics '*839*' and '*878*'.

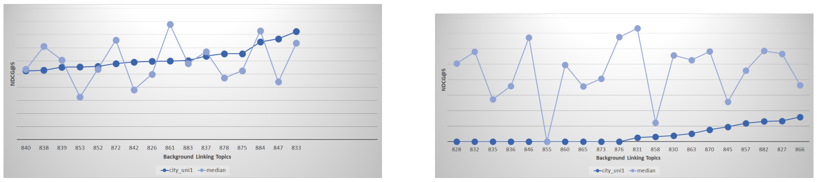


(a) Topics for which our methods achieved the highest nDCG@5

(b) Topics for which our methods achieved the lowest nDCG@5

Fig. 1: A Comparison over the topics for which our methods achieved the highest and lowest nDCG@5 for Entity Ranking task

For the Background Linking task, our method $city\_uni1$ achieves the highest gain comparing to the median over the topic *'847'* and *'853'* with more than 70% of gain. As shown in fig 2(b), the method was less accurate in many topics including $828, 832, 835$, etc. In more in depth analysis, we observe that our methods suffer from biggest losses in topic that relatively has short title or title that encompass few Named Entities in them. This fact makes it harder for the method to create a pool of relevance feedback based on the topic title only.



(a) Topics for which our methods achieved the highest nDCG@5

(b) Topics for which our methods achieved the lowest nDCG@5

Fig. 2: A Comparison over the topics for which $city\_uni1$ method achieved the highest and lowest nDCG@5 for Background Linking task

All our proposed methods for both tasks are heavily based on the basic assumption that relevance assessments are available for a given topic, and that these relevant documents can be used to extract a set of useful Named Entities to represent the topic. However, given the evaluation results and the analysis we made, we can argue that our methods could perform better in the case of having a real pseudo-relevance feedback. For this reason, we are planning to conduct further experiments with this year TREC results to re-investigate the accuracy of our three methods comparing to some other baseline methods.

6 Conclusion

In this year's TREC News Track, we investigated using Named Entities in different ways to find background news articles and to rank the mentioned list of entities. For the Background Linking task, we experimented with a simple heuristic learning method that has been optimized on an initial set of entities to retrieve background documents. Most notably, our enhancement for efficiency shows that the computational complexity of the Hill Climber approach can be reduced with using NEs instead of terms.

We approached the Entity Ranking task using two different methods; Named Entities were used in both and the Wikipedia Dump was the key success for the $City\_ER2$ run.

To this end, we adapted the state-of-the-art BM25 algorithm along with our previous work in TREC. Overall our adaptation is promising, as all our runs yield reasonable results.

Our methods are heavily based on the basic assumption that relevance assessments are available for a given topic. An in depth analysis of our adapted methods and results has shown that using the topic's title to create a pool of relevance feedback may arguably sacrifices some retrieval effectiveness. To overcome this problem, we are planning a future experimentation using the real pseudo-relevance feedback provided by News Track for 2019 topics.

Bibliography

[1] M Beaulieu, M Gatford, Xiangji Huang, S Robertson, S Walker, and P Williams. Okapi at trec-5. *NIST SPECIAL PUBLICATION SP*, pages 143–166, 1997.

[2] Christopher Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Gregory N Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd International Conference on Machine learning (ICML-05)*, pages 89–96, 2005.

[3] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 363–370. Association for Computational Linguistics, 2005.

[4] Manish Gupta, Michael Bendersky, et al. Information retrieval with verbose queries. *Foundations and Trends® in Information Retrieval*, 9(3-4):209–354, 2015.

[5] Ben Hachey, Will Radford, Joel Nothman, Matthew Honnibal, and James R Curran. Evaluating entity linking with wikipedia. *Artificial intelligence*, 194:130–150, 2013.

[6] Donna K Harman. *Overview of the third text retrieval conference (TREC-3)*. Number 500. DIANE Publishing, 1995.

[7] David A Hull et al. The trec-7 filtering track: description and analysis. *NIST SPECIAL PUBLICATION SP*, pages 45–68, 1998.

[8] Kuang Lu and Hui Fang. Paragraph as lead - finding background documents for news articles. In *Proceedings of the Twenty-Seventh Text REtrieval Conference, TREC 2018, Gaithersburg, Maryland, USA, November 14-16, 2018*, 2018.

[9] Andrew MacFarlane, Stephen E Robertson, and Julie A McCann. Pliers at vlc2. *NIST Special Publication 500-242*, 1999.

[10] Andrew MacFarlane, Stephen E Robertson, and Julie A McCann. Pliers at trec8. *NIST Special Publication 500-246*, 2000.

[11] Andrew MacFarlane, Andrew Secker, Peter May, and Jonathan Timmis. An experimental comparison of a genetic algorithm and a hill-climber for term selection. *Journal of documentation*, 66(4):513–531, 2010.

[12] Alberto Paro. *ElasticSearch cookbook*. Packt Publishing Ltd, 2015.

[13] Samira Pouyanfar, Saad Sadiq, Yilin Yan, Haiman Tian, Yudong Tao, Maria Presa Reyes, Mei-Ling Shyu, Shu-Ching Chen, and SS Iyengar. A survey on deep learning: Algorithms, techniques, and applications. *ACM Computing Surveys (CSUR)*, 51(5):92, 2019.

[14] Stephen E Robertson. On term selection for query expansion. *Journal of documentation*, 46(4):359–364, 1990.

[15] Stephen E Robertson, Steve Walker, MM Beaulieu, Mike Gatford, and Alison Payne. Okapi at trec-4. In *Proceedings of the fourth text retrieval conference*, volume 500, pages 73–97. NIST Special Publication Gaithersburg, MD, 1996.

[16] Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. Okapi at trec-3. *Nist Special Publication Sp*, 109:109, 1995.

[17] Olga Vechtomova. *Query Expansion for Information Retrieval*, pages 2254–2257. Springer US, Boston, MA, 2009.

[18] Li Zhang, Yue Pan, and Tong Zhang. Focused named entity recognition using machine learning. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 281–288. ACM, 2004.