FEUP at TREC 2018 Common Core Track

Reranking for Diversity using Hypergraph-of-Entity and Document Profiling

José Devezas
INESC TEC & FEUP, University of Porto
Porto, Portugal
jld@fe.up.pt

Antonio Guillén
DLSI, Universidad de Alicante
Alicante, Spain
aguillen@dlsi.ua.es

Yoan Gutiérrez
DLSI, Universidad de Alicante
Alicante, Spain
ygutierrez@dlsi.ua.es

Rafael Muñoz
DLSI, Universidad de Alicante
Alicante, Spain
rafael@dlsi.ua.es

Sérgio Nunes
INESC TEC & FEUP, University of Porto
Porto, Portugal
ssn@fe.up.pt

ABSTRACT

We describe our participation in the TREC 2018 Common Core track, where we experimented with hyperedge-based document ranking, over the hypergraph-of-entity. We compared a text-only implementation (*feup-run1*) with a different implementation that also included entities and triples from DBpedia (*feup-run2*). We also experimented with reranking for diversity, based on the maximal marginal relevance and document profiling, in order to find a balance between relevance and the dissimilarity of neighboring documents. This resulted in six additional runs (3 to 8), using *feup-run1* and *feup-run2* as the base runs for reranking. We then assessed the impact in effectiveness, along with the changes in diversity, particularly over the top-ranked documents. We evaluated retrieval effectiveness based on the mean average precision, over the relevance judgments provided by TREC. We also proposed a weighted diversity metric, based on the cosine distance between each document and all others, within results for the same topic. Documents with a lower rank were assigned a higher weight, more strongly contributing to the weighted diversity. Our best results were for *feup-run1* and *feup-run7*, both with a MAP score of 0.0070 and a P@10 of 0.0680, as well as a weighted diversity of 0.1197 and 0.1218, respectively.

1 INTRODUCTION

Graphs are general data structures capable of capturing discourse properties from text [3], as well as knowledge from entities and their relations [2]. Graphs can support multiple tasks, from query understanding [15] to entity disambiguation [16] and document retrieval [18]. Haentjens Dekker and Birnbaum [14] go even further as to represent text as a hypergraph. Hypergraphs are a generalization of graphs where edges, or rather hyperedges, can contain an arbitrary number of nodes. Undirected hyperedges can be represented by a set of nodes (e.g., the terms in a sentence), while directed hyperedges can be represented by a tail set of nodes and a head set of nodes[1] (e.g., an e-mail to multiple recipients). Hypergraphs can be represented as an equivalent bipartite incidence graph, where each original node connects to a new node for each of its hyperedges. While such conversion is possible, it is not always ideal. This is true for instance for the study of overlapping or hierarchical relations. When broken down into multiple edges, they become harder to identify or read. In this work, we use the hypergraph-of-entity as an indexing data structure for the TREC Washington Post Corpus, where we represent terms, entities and their relations. For the base models, we explore both a text-only representation (*feup-run1*) and a joint representation of text and knowledge from DBpedia (*feup-run2*).

Document profiling proposal consists in generating a profile from documents able to represent different features extracted through human language technologies (HLT). These HLT cover areas like Sentiment Analysis, Topic and Keyword Extraction, Named Entity Recognition, Readability Analysis, etc. Gulla et al. [11] introduce this idea but in terms of user profiles. These profiles are generated through the interaction and user behavior collected on the Internet for news recommendation. Usbeck [19] address the importance of using meta-data information in Information Retrieval because it emphasizes search quality. Therefore, our idea is to apply the document profiles over the top-ranked documents to generate some diversity improving variety without compromising relevance.

We used maximal marginal relevance (MMR) [4] to rerank documents from the base runs, *feup-run1* and *feup-run2*, using $Sim_1(D_i, Q)$ as a normalization of the score already provided in the base runs and experimenting with multiple $Sim_2(D_i, D_j)$ based on document profiles with different features. We then assessed the effectiveness of each run based on the relevance judgments provided by TREC and calculated a weighted diversity score to understand how the reranking affected the diversification of the top results.

The remainder of this article is organized as follows. In Section 2, we present the TREC Washington Post Corpus, characterizing the temporal distribution and the length distribution of the documents. In Section 3, we detail the hypergraph-of-entity as a representation and retrieval model, describing our ranking function based on random walks, as well as the configuration for the two base runs. In Section 4, we describe the tools, algorithms and features used to build each document profile, as well as the reranking process based on MMR and the multiple configurations that resulted in six additional runs. Finally, in Section 5, we assess the retrieval effectiveness, using MAP and P@10, and measure the aggregated diversity for each run, aiming to understand whether it is possible

[1] The analogy is to an arrow, not to a list. This is why the tail set contains the source nodes and the [arrow]head set contains the target nodes.

TREC 2018, November 2018, Gaithersburg, Maryland, USA
2018.

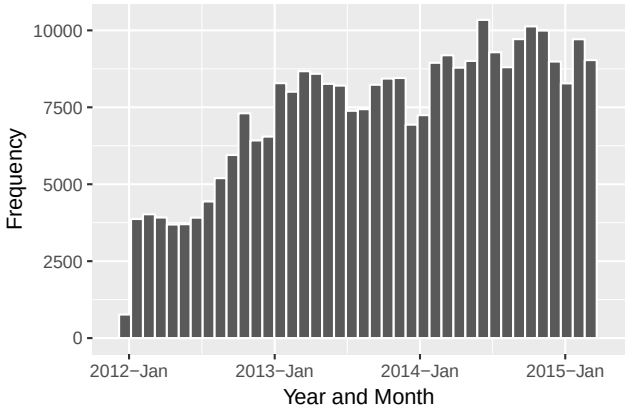

Figure 1: Temporal distribution of documents.

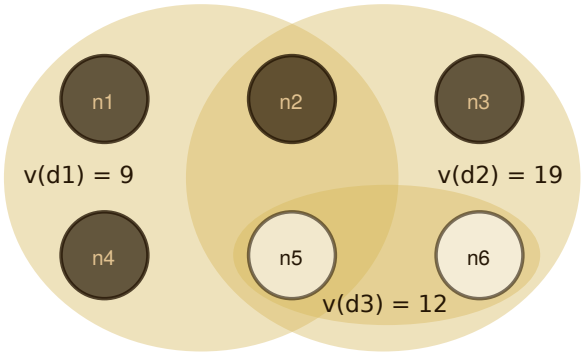

Figure 2: Hypergraph-of-entity: ranking *document* hyperedges using $RWS(\ell = 2, r = 10)$ for seed nodes $n5$ and $n6$.

to diversify results without impacting effectiveness. In Section 6 we present some final remarks and conclusions.

2 TREC WASHINGTON POST CORPUS

We used TREC Washington Post Corpus[2] version 2 provided by TREC's Common Core Track. This corpus contains 595,037 documents without 13,143 repeated entries removed from Corpus version 1 (repeated entries are 8,849 distinct document ids). Specifically, 236,649 news articles and 358,388 blog posts. These documents have been published between January 2012 and August 2017. Figure 1 shows the number of articles published month by month.

The documents are stored in JSON format and include these fields:

- *id*: document identification.
- *title*: a text field for the document main title.
- *author/byline*: author(s) of the publication.
- *date of publication (published_date)*: publication date in timestamp format.
- *kicker*: a section header indicating the publication category.
- *content*: article text broken into paragraphs.
- *type*: blog or article.
- *article_url*: URL of the publication.
- *source*: The Washington Post.

Article text is split into paragraphs, links to embedded images and multimedia. Each document is identified by the *id* field. In total, the collection file has 1.5GB (compressed) or 6.9GB (uncompressed). Paragraphs are stored in a list of contents as sub-type *paragraph*, including HTML tags (removed for text processing). We only used the three first paragraphs with our approach.

3 HYPERGRAPH-OF-ENTITY

We used a hypergraph-based data structure, evolved from the graph-of-entity [6], to index the test collection. The hypergraph-of-entity is able to represent textual documents, with associated named entities and their relations, through two types of nodes — *term* nodes and *entity* nodes — and three types of hyperedges — undirected *document* hyperedges, which aggregate all terms and entities within a document, undirected *related_to* hyperedges, which link sets of related entities, and directed *contained_in* hyperedges, which link a set of terms to their corresponding entity. Last year, in TREC 2017 OpenSearch Track, the FEUP team had explored a similar representation approach with the graph-of-entity [5]. Ranking in the graph-of-entity is based on the entity weight, while ranking in the hypergraph-of-entity is based on the random walk score. Both models use a common seed node selection process, where a keyword query is mapped to a set of term nodes, which are, in turn, expanded to adjacent entity nodes. These neighboring entity nodes then become the seed nodes, unless no entity node is linked to a given term node, which then becomes its own seed node. Calculating the random walk score (RWS) requires launching multiple random walkers from the identified seed nodes, with a given length ℓ and a number of iterations r. Each random step consists of randomly selecting a hyperedge and then randomly selecting a node from that hyperedge. The number of visits to *document* hyperedges is then collected and used to obtain a final score and ranking.

Figure 2 illustrates the output of such a random walk, showing three documents represented by their hyperedges, $d1$, $d2$ and $d3$, and six abstract nodes (i.e., they can either represent *term* or *entity* nodes), two of which, $n5$ and $n6$, are identified as seed nodes. We calculated $RWS(\ell = 2, r = 10)$ by launching 10 random walks of length 2 from each of the seed nodes, aggregating the number of visits per hyperedge as $v(d)$. As we can see, both $d1$ and $d2$ contain four nodes, overlapping on $n2$ and $n5$, and $d2$ subsumes $d3$, that is, it is more general than $d3$, containing all of its nodes (and more). One of the possible results of the nondeterministic execution of the random walk score is show in the figure. Larger values of r (usually $r > 1000$) will result in a higher ranking consistency, however, for such a small hypergraph, $r = 10$ is sufficient to converge to the shown ranking: $d2$, $d3$, $d1$.

3.1 Base runs

We prepared and submitted two base runs to TREC Common Core track, which were then used to generate an additional six runs based on reranking for diversity over different document profiles (see Section 4). The first run (*feup-run1*) was based on a text-only version

[2] https://trec.nist.gov/data/wapost/

Table 1: Statistics for the hypergraphs-of-entity used in *feup-run1* and *feup-run2*.

Version	Statistic	Value
Text-Only	*term* nodes	886,298
	undirected *document* hyperedges	595,037
DBpedia	*term* nodes	886,298
	entity nodes	276,735
	Total nodes	1,163,033
	undirected *document* hyperedges	595,037
	undirected *related_to* hyperedges	595,037
	Total undirected hyperedges	1,190,074
	directed *contained_in* hyperedges	266,962
	Total hyperedges	1,457,036

of the hypergraph-of-entity, simply consisting of *term* nodes and *document* hyperedges. The second run (*feup-run2*) was an extension of the text-only version, where we added DBpedia [2] entities and the respective triples for each entity, relying on *entity* nodes, *related_to* hyperedges and *contained_in* hyperedges for the representation. The information extraction process of *feup-run2* was limited to the first three paragraphs of each document, simply due to resource constraints — the current implementation requires the hypergraph-of-entity to be fully loaded into RAM and we were, at this point, unable to consider all triples for all extracted entities using the available 30 GB of RAM. Named entity recognition (NER) was carried based on the Aho-Corasick string-searching algorithm [1] over a combined list of all '@en' *rdfs:label* for *dbo:Person*, *dbo:Organisation* and *dbo:Place* entities. An HTTP endpoint implementing this strategy is available in Army ANT[3]. It can be run by first setting up `defaults/service/ner/entity_list` in `config.yaml` and launching server, and then sending a `POST` request to `http://localhost:8080/service/ner`, using the field `text` to get the list of identified entities. This NER strategy was chosen so that we could more efficiently match a finite list of labels with each document.

Table 1 shows several statistics for the hypergraphs-of-entity used in the base runs 1 and 2. No stemming or lemmatization was applied to either version, resulting in a vocabulary of over 800 thousand terms. When extended with DBpedia information, over 200 thousand people, organizations and places were extracted as entities. As expected, there were as many *document* hyperedges as documents in the collection. We also included *related_to* hyperedges to model co-occurrence of entities in documents, but this should be improved in the future to better take advantage of triples associated with each document and its entities. Finally, over 200 thousand relations were established between sets of terms and their corresponding entity — this was based on term matching with the entity name, but it could easily use another type of term–entity association instead, or be extended to consider other languages, for cross-language retrieval.

[3]https://github.com/feup-infolab/army-ant/tree/trec-2018

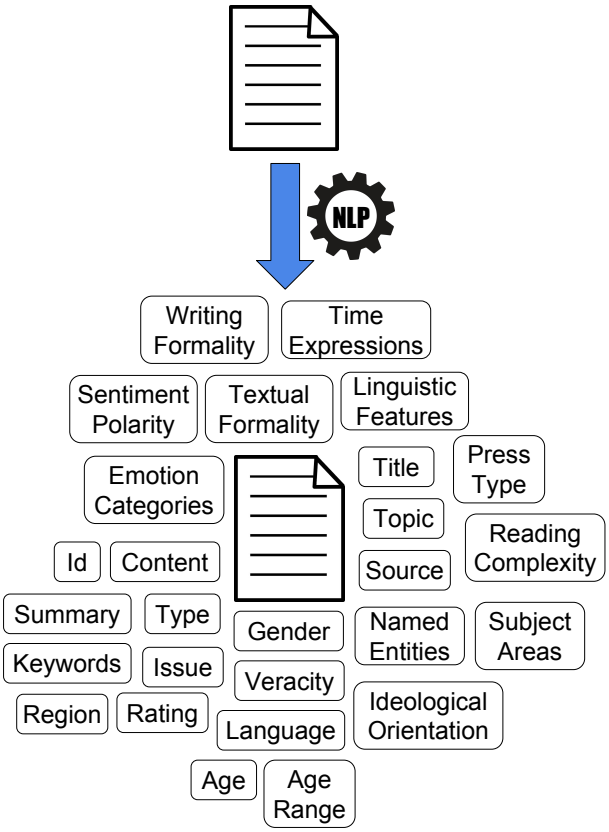

Figure 3: Document profiling concept.

Table 2: List of features used in document profile.

Feature	Code
Named Entities	NE
Sentiment Analysis	SA
Emotion Categories	EC
Reading Complexity	RC
Keywords	KW

4 DOCUMENT PROFILING

This proposal consists of designing a document profile able to represent different features extracted from text documents using HLT tools [10]. Figure 3 provides an overview of this concept. As we can be seen, there are many features that can only be extracted automatically from documents using these technologies. For this work, only a small set of features is used because the amount of text from corpus requires technologies with a good performance in huge text collections. Table 2 shows an overview of these features used for document profiling of TREC's documents: name of the feature and code. An extended detail of each feature is described below.

Named Entities

Initially, it was used Stanford NER [9] that uses CRF-based information extraction system with long-distance dependency models. This technique has a high precision and reliability, but is very slow in a large collection of data. For this reason, it was used Aho-Corasick algorithm [1] that uses a string-searching and dictionary-matching algorithm using a list of entities obtained from DBpedia.

Sentiment Analysis

This feature consists in analyzing English texts to detect sentiment polarities. The system is based on supervised machine learning and text categorization techniques, and ranking skip-gram techniques [12]. The model has been trained by using Twitter posts and movies reviews. The polarity classifications can be positive, negative and neutral without intensity weight.

Emotion Categories

This is a part of Sentiment Analysis but obtaining concrete emotions from the text. In this case it was used GPLSI Emotion Analysis V1.0 API [13]. This technology uses Weka on a Support Vector Machines (SVM) algorithm. The SVM features are extracted by using skip-grams, and training dataset used has been obtained from Potter, Grimms and H. C. Andersen. Each value of emotion has a weight or intensity, concretely a double value between 0 and 1.

Reading Complexity

In order to estimate the text understanding complexity it was applied Flesch–Kincaid score [8] through textstat[4]. The output is a result between 0 and 100, that is categorized as defined:

- 5th_grade: 90 to 100: (Very Easy)
- 6th_grade: 80 to 90 (Easy)
- 7th_grade: 70 to 80 (Fairly Easy)
- 8th_and_9th_grade: 60 to 70 (Standard)
- 10th_to_12th_grade: 50 to 60 (Fairly Difficult or high school)
- college: 30 to 50 (Difficult)
- college_graduate: 0 to 30 (Very Confusing)

Keywords

For this feature, it was used RAKE tool [17] to extract automatic keywords from documents. The method consists of parsing text forming arrays of continuous words without stop-words. A candidate keyword is identified and included as individual words in the graph of word co-occurrences metric, being *deg(w)/freq(w)* used to calculate individual word scores. The weight for each candidate keyword is computed as the sum of its member word scores.

4.1 Reranking runs

Runs 3 to 8 are based on *feup-run1* or *feup-run2*, applying document profile on corpus' documents. Table 4 shows information about MMR's λ parameter used in each run, and the list of the features included to calculate MMR. Diversity is controlled by the λ parameter of MMR algorithm. For a normal diversity is used 0.85, and for strong diversity is used 0.50. A lower than 0.50 could cause an excess of diversity.

[4]https://pypi.org/project/textstat

Table 3: Properties of features-based runs.

Run	Based	λ	Features
feup-run3	feup-run1	0.85	KW NE SA RC EC
feup-run4	feup-run2	0.85	SA RC
feup-run5	feup-run1	0.50	KW NE SA RC EC
feup-run6	feup-run2	0.85	SA RC EC
feup-run7	feup-run1	0.85	SA RC
feup-run8	feup-run1	0.85	NE SA RC EC

Table 4: Evaluation of runs regarding retrieval effectiveness and results diversity.

Run	MAP	P@10	*wd* (base)	*wd* (rerank)	p-value
feup-run1	0.0070	0.0680	-	-	-
feup-run2	0.0051	0.0240	-	-	-
feup-run3	0.0069	0.0660	0.8249	0.8257	0.8632
feup-run4	0.0033	0.0260	0.1106	0.1288	0.0007
feup-run5	0.0068	0.0580	0.8249	0.8280	0.6221
feup-run6	0.0028	0.0200	0.4372	0.4770	2.42e-05
feup-run7	0.0070	0.0680	0.1197	0.1218	0.8496
feup-run8	0.0069	0.0640	0.8603	0.8615	0.8173

KW and *EC* features could have a huge quantity of possibles values. Is considered a way to reduce this by binary conversion of weight and removing some values fewer representatives (low weight emotions, keywords with a large number of words). One way of reduction is converting weight values to binary values. This represents the usage or not of numerical weights in *KW* and *EC* features. Binary means that weights are not taken into account in the MMR algorithm, only 1 is considered if a value of a feature is defined and 0 it is not defined. Otherwise, is considered the weight for MMR algorithm.

Another way of reduction is removing values in *KW* and *EC* features. This represents the limitation of certain values by weight or the number of words in keywords. Keywords with a weight lower than 5.0 or more than 2 terms, and Emotion Categories with a weight lower or equal to 0.1 were discarded for runs 3, 4, 5, 6 and 7. The only run that is considered all weights in *EC* is *feup-run8* but not included *KW* features. Also, runs based in *feup-run2* have only included a top 100 entries per topic.

5 EVALUATION

We assessed retrieval effectiveness based on the relevance judgments provided by TREC and based on the 50 test topics for 2018. The provided test topics consisted of 25 topics from the 2017 Common Core track, as well as 25 new topics prepared by NIST assessors. All experiments were run over the TREC Washington Post Corpus using the title of the topic as a keyword query. Each submitted run was assigned a priority, and participants were assured that at least two runs per team would be judged, at the very least considering the top 10 documents per topic. We assigned the top priority to the base runs (*feup-run1* and *feup-run2*), simply because the remaining runs

resulted from a reranking of the same exact set of documents, with a high probability of sharing a similar set of top 10 documents, given the MMR-based rerank for diversity. In the following two sections, we analyze the evaluation results from TREC, and then measure diversity for different rerankings based on a proposed a weighted diversity metric, discussing the relation between effectiveness and diversity.

5.1 Retrieval Effectiveness

The command `trec_eval -c -q -M1000` was used by TREC to calculate multiple effectiveness metrics, where `-c` ensures that the average is done over the complete set of queries in the relevance judgments, `-q` includes per-topic evaluations, and `-M1000` considers only a maximum of 1,000 retrieved documents. Out of the provided metrics, we selected the mean average precision (MAP) as the overall effectiveness indicator and the precision at a cutoff of 10 (P@10) as a complement.

As we can see in Table 4, the best runs were *feup-run1* and *feup-run7*, both with a MAP of 0.0070 and a P@10 of 0.0680. Additionally, the difference between MAP scores for different runs is not statistically significant. Additionally, it is clear that, overall, precision is quite low, with a few MAP scores achieving a maximum of 0.1429 for topic 810 and runs 1, 3, 5, 7 and 8, and the next best MAP scores of 0.1312, 0.0815 and 0.0715 for topic 822 and runs 2, 4 and 6, respectively. Following topics 810 and 822 there is topic 823 with lower MAP scores of 0.0294 for runs 1 and 7, 0.0293 for run 3 and 0.0289 for run 8. Over half of the topics for all runs resulted in a MAP score of zero. This includes for instance topic 427 for run 3 and 4, topic 341 for run 1 and topic 819 for run 3.

Table 5 shows the keyword queries corresponding to the best and worst topics according to MAP. The best results, for topic 810, were consistently based on the text-only version of the hypergraph-of-entity. Also, as expected, whenever we obtained a good MAP score for a base run, the derived runs also resulted in a similar MAP score. According to this small sample, there is no clear advantage of using the DBpedia version over the text-only version of the hypergraph-of-entity, or vice-versa. This might become clearer with a better usage of the triples associated with the documents, when available. The table also shows the number of relevant documents retrieved (Rel. Retr.) versus the number of relevant documents in the collection (Rel.), as well as the number of partial node matches in both versions of the hypergraph-of-entity, so that we better understand the amount of information potentially covering each term. As we can see, with the exception of topic 822 (`[ Sony cyberattack ]`) very few relevant documents were retrieved. Additionally, the highest MAP score is justified by a low overall number of relevant documents. When looking at the number of matching nodes per query term, we found that this is also not an indicator of effectiveness, as the worst topics can either have a high or low number of matching nodes.

A manual investigation, however, provided a few insights as to what might have caused such a low overall precision. First, we found a tokenization issue where stopwords weren't sometimes split from other words (e.g., "and.hacking", "economics.and"), needlessly extending the vocabulary and discarding paths for random walks. This was also the first time we indexed a collection of news articles, after having worked with encyclopedic content, and we found that random walks of length $\ell = 2$ would easily reach unrelated topics. For example, when departing from "diabetes", we would step into a article entitled "Can running help autistic children?" and then randomly into the term "megan", which would lead to "Home sales in Loudoun and Fauquier counties". The constraints provided by the hypergraph-of-entity are still not enough, in particular to support search using random walks over a collection of news articles. Several approaches might be taken to improve this, namely introducing *sentence*, *paragraph* or *passage* hyperedges in order to avoid taking steps into unrelated directions (such as "megan"). Obviously, despite document scoring depending on $r = 1000$ random walks for each seed node (frequently multiple entities for a single term), allowing such unrelated walks is still detrimental to the overall ranking. Finally, the hypergraph-of-entity does not support any type of document length normalization, which is also affecting the quality of random walks. We also did not use any stemming or lemmatization, since we wanted to leave room for the exploration of syntactic relations, which could only be extracted and modeled based on complete sentences. We propose that such issues, along with the lack of proper usage of available triples are reasons for the low MAP scores that we obtained.

5.2 Document diversity

Besides retrieval effectiveness, we also measured the diversity of the ranked documents, taking order into account. The idea was that a ranker is better if it returns a higher number of relevant documents in the top ranks, but also if it ensures the diversification of rank-adjacent documents. In Devezas and Nunes [7], diversity was measured based on the cosine distance between each Twitter timeline and all other timelines, with a timeline being represented by a vector of term frequencies. The diversity was then considered high whenever the median cosine distance, within a set of timelines, was also high. In this work, we follow a similar approach, but use a weighted mean to assign exponentially decreasing weights (Eq. 1), from the top to the bottom ranked documents.

$$w_i = (i \cdot ||w||_1)^{-1} \tag{1}$$

For the sake of consistency with the MMR notation, we refer to the cosine similarity between two documents D_i and D_j as $Sim_2(D_i, D_j)$, where D_i and D_j are represented by vectors of features, according to the chosen document profile. Let wd_{topic} be the weighted diversity of a given topic with N retrieved ranked documents (Eq. 2).

$$wd_{topic} = \sum_{i=1}^{N} \left(w_i \cdot \frac{1}{N} \sum_{j=1, j \neq i}^{N} Sim_2(D_i, D_j) \right) \tag{2}$$

As we can see, we calculate the average cosine distance between a document D_i and all other ranked documents D_j and then calculate the weighted average over the resulting values. Assigning higher weights to top-ranked documents enables us to compare the impact in diversity for different rankings of the same set of documents. Finally, we aggregate the weighted diversity per topic (wd_{topic}) for a whole run by calculating the average over all topics. The script

Table 5: Characteristics of best and worst retrieved topics.

Topic	Best MAP	Query	Rel. Retr.	Rel.	Node Matches	
					Text-Only	DBpedia
Best						
810	0.1429	`[ diabetes and toxic chemicals ]`	1	7	15-NA-52-10	21-NA-62-15
822	0.1312	`[ Sony cyberattack ]`	34	43	24-6	78-6
823	0.0294	`[ control of MRSA ]`	6	58	93-NA-2	164-NA-2
Worst						
341	0.0001	`[ Airport Security ]`	0	124	32-67	621-238
427	0.0000	`[ UV damage, eyes ]`	0	28	NA-10-109	NA-17-295
819	0.0000	`[ U.S. age demographics ]`	0	15	7,949-3,481-2	12,850-5,896-2

that computes this score can be found at FEUP InfoLab's GitHub under the TREC 2018 repository[5].

In the right side of Table 4, we find the weighted diversity for the base run and the reranked run, measured using the same document profile. We also provide the p-value based on the Mann-Whitney U test. As we can see, the weighted diversity is, as expected, consistently higher for the reranked runs, however this is only statistically significative for *feup-run4* and *feup-run6*, which are both based on *feup-run2* and both are based on document profiles that use Sentiment Analysis and Reading Complexity, with *feup-run6* also using Emotion Categories.

6 CONCLUSIONS

We submitted eight runs to TREC 2018 Common Core track. We explored two hypergraph-based models, one of them using a text-only approach and the other extended with external knowledge based on DBpedia entities and triples. We then applied several reranking strategies based on different α values for the maximal marginal relevance and different document profiles, with the goal of diversifying search results without impacting effectiveness. Although, overall, we did not achieve a high retrieval effectiveness, we identified several issues with the hypergraph-of-entity that should be corrected in the future to hopefully improve effectiveness. Regarding the reranking runs, we proposed a weighted metric to assess diversity, even for different rankings of the same set of documents, where we assigned a higher relevance to the top ranks. The hypothesis we wanted to test was whether reranking for diversity based on document profiles, considering metadata like sentiment, emotion or readability, would result in a similar effectiveness, while improving the diversity for top documents. While further investigation is still required, we already found leads to indicate that this is possible. In particular, there was a statistically significant difference in weigthed diversity between the base models and the derived models in *feup-run4* and *feup-run6*, both based on *feup-run2*, the DBpedia version of the hypergraph-of-entity.

ACKNOWLEDGMENTS

José Devezas is supported by research grant PD/BD/128160/2016, provided by the Portuguese national funding agency for science, research and technology, Fundação para a Ciência e a Tecnologia (FCT), within the scope of Operational Program Human Capital (POCH), supported by the European Social Fund and by national funds from MCTES. Antonio Guillén is supported by the University of Alicante through a fellowship from Formación de Profesorado Universitario program (UAFPU2015-5999), and partially funded by the University of Alicante, Generalitat Valenciana, Spanish Government, Ministerio de Educación, Cultura y Deporte and Ayudas Fundación BBVA a equipos de investigación científica 2016 through the projects TIN2015-65100-R, TIN2015-65136-C2-2-R, GRE16-01: "Plataforma inteligente para recuperación, análisis y representación de la información generada por usuarios en Internet" and Análisis de Sentimientos Aplicado a la Prevención del Suicidio en las Redes Sociales (ASAP).

[5] https://github.com/feup-infolab/trec-2018/blob/master/eval_diversity.R

REFERENCES

[1] Alfred V. Aho and Margaret J. Corasick. 1975. Efficient String Matching: An Aid to Bibliographic Search. *Commun. ACM* 18, 6 (1975), 333–340. https://doi.org/10.1145/360825.360855

[2] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary G. Ives. 2007. DBpedia: A Nucleus for a Web of Open Data. In *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007.* 722–735. https://doi.org/10.1007/978-3-540-76298-0_52

[3] Roi Blanco and Christina Lioma. 2012. Graph-based term weighting for information retrieval. *Information Retrieval* 15, 1 (2012), 54–92. https://doi.org/10.1007/s10791-011-9172-x

[4] Jaime G. Carbonell and Jade Goldstein. 1998. The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries. In *SIGIR '98: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 24-28 1998, Melbourne, Australia.* 335–336. https://doi.org/10.1145/290941.291025

[5] José Devezas, Carla Teixeira Lopes, and Sérgio Nunes. 2017. FEUP at TREC 2017 OpenSearch Track: Graph-Based Models for Entity-Oriented Search. In *The Twenty-Sixth Text REtrieval Conference Proceedings (TREC 2017).* Gaithersburg, MD, USA.

[6] José Devezas and Sérgio Nunes. 2017. Graph-Based Entity-Oriented Search: Imitating the Human Process of Seeking and Cross Referencing Information. *ERCIM News. Special Issue: Digital Humanities* 111 (Oct. 2017), 13–14.

[7] José Luís Devezas and Sérgio Nunes. 2018. Social Media and Information Consumption Diversity. In *Proceedings of the Second International Workshop on Recent Trends in News Information Retrieval co-located with 40th European Conference on Information Retrieval (ECIR 2018), Grenoble, France, March 26, 2018.* 18–23. http://ceur-ws.org/Vol-2079/paper5.pdf

[8] Flesch Reading Ease. 2009. Flesch–Kincaid readability test. (2009).

[9] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL '05).* Association for Computational Linguistics, Stroudsburg, PA, USA, 363–370. https://doi.org/10.3115/1219840.1219885

[10] Antonio Guillén, Yoan Gutiérrez, and Rafael Muñoz. 2017. Natural Language Processing Technologies for Document Profiling. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017, Varna, Bulgaria, September 2 - 8, 2017*. 284–290. https://doi.org/10.26615/978-954-452-049-6_039

[11] Jon Atle Gulla, Arne Dag Fidjestøl, Xiaomeng Su, and Humberto Castejon. 2014. Implicit User Profiling in News Recommender Systems. *International Conference on Web Information Systems and Technologies* (2014). https://doi.org/10.5220/0004860801850192

[12] Yoan Gutierrez, David Tomas, and Javi Fernandez. 2016. Benefits of using ranking skip-gram techniques for opinion mining approaches. In *eChallenges e-2015 Conference Proceedings*. https://doi.org/10.1109/eCHALLENGES.2015.7441056

[13] Yoan Gutiérrez, David Tomás, Isabel Moreno, and Javier Fernández Martínez. 2017-06-05. GPLSI Emotion Analysis V1.0: Análisis de emociones en textos. (2017-06-05).

[14] Ronald Haentjens Dekker and David J. Birnbaum. 2017. It's more than just overlap: Text As Graph. In *Proceedings of Balisage: The Markup Conference 2017*, Vol. 19. https://doi.org/10.4242/BalisageVol19.Dekker01

[15] Jian Hu, Gang Wang, Fred Lochovsky, Jian-tao Sun, and Zheng Chen. 2009. Understanding user's query intent with wikipedia. In *Proceedings of the 18th international conference on World wide web*. ACM, 471–480.

[16] Andrea Moro, Alessandro Raganato, and Roberto Navigli. 2014. Entity Linking meets Word Sense Disambiguation: a Unified Approach. *Transactions of the Association for Computational Linguistics* 2 (2014), 231–244. https://tacl2013.cs.columbia.edu/ojs/index.php/tacl/article/view/291

[17] Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. 2010. Automatic Keyword Extraction from Individual Documents. In *Text Mining: Applications and Theory*. https://doi.org/10.1002/9780470689646.ch1

[18] François Rousseau and Michalis Vazirgiannis. 2013. Graph-of-word and TW-IDF: new approach to ad hoc IR. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*. ACM, 59–68.

[19] Ricardo Usbeck. 2014. Combining Linked Data and Statistical Information Retrieval. (2014).