Detecting and Ranking Conceptual Links between Texts Using a Knowledge Base

Martin Tutek[1], Goran Glavaš[1,2], Jan Šnajder[1],
Nataša Milić-Frayling[3], Bojana Dalbelo Bašić[1]
[1] Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia
[2] Data and Web Science Group, University of Mannheim, Germany
[3] School of Computer Science, University of Nottingham, United Kingdom
{martin.tutek, goran.glavas, jan.snajder, bojana.dalbelo}@fer.hr
natasa.milic-frayling@nottingham.ac.uk

ABSTRACT

Recent research has explored the use of Knowledge Bases (KBs) to represent documents as subgraphs of a KB concept graph and define metrics to characterize *semantic relatedness* of documents in terms of properties of the document concept graphs. However, none of the studies so far have examined to what degree such metrics capture a *user-perceived* relatedness of documents. Considering the users' explanations of how pairs of documents are related, the aim is to identify concepts in a KB graph that express the same notion of document relatedness. Our algorithm generates paths through the KB graph that originate from the terms in two documents. KB concepts where these paths intersect capture the semantic relatedness of the two starting terms and therefore the two documents. We consider how such intersecting concepts relate to the concepts in the users' explanations. The higher the users' concepts appear in the ranked list of intersecting concepts, the better the method in capturing the users' notion of document relatedness. Our experiments show that our approach outperforms a simpler graph method that uses properties of the concept nodes alone.

Keywords

Content analysis; semantic relatedness, knowledge base graph

1. INTRODUCTION

Typical information retrieval (IR) tasks, such as search, query expansion, and recommendation, rely upon techniques that involve (1) a representation of document content and (2) operators applied to that representation to measure document relevance to a user's query or similarity of compared documents. Most traditional IR approaches use a 'bag-of-words' document representation and relevance metrics based on lexical overlap [9, 7, 8]. As such, they are not well equipped to support more complex information needs requiring generalization or enumeration of specific information items. For example, the following two text snippets have no lexical overlap but are both relevant to the query "recent natural disasters in Asia":

> *"In July 2013, much of southwest China experienced heavy rainfall that led to widespread flooding…"*
>
> *"It was the most powerful earthquake ever recorded to have hit Japan…"*

Methods such as Latent Semantic Indexing [2] and Latent Dirichlet Allocation [1] aim to capture the underlying semantic meaning and establish relatedness of documents beyond lexical overlap. Unfortunately, they cannot easily provide a user-interpretable explanation of uncovered semantic relations.

Recently we have seen efforts to explicate the semantic relation among documents by using knowledge bases (KBs) to characterize relations among document terms. By mapping document terms onto KB concepts, each document is represented as a *concept subgraph* of a KB graph [10, 6]. This representation presents opportunities for formalizing retrieval operations as operators over graphs that leverage both explicit concept links and statistical properties of terms in the document corpus. An essential step in using KB representations is to define metrics to characterize *relatedness* of two documents. That can be achieved by first defining *pairwise relatedness of terms* and then deriving *relatedness of documents* containing the terms.

While there are many principled ways of using the concept graphs to define semantic similarity of documents, there is a fundamental question of whether the relatedness metrics agree with the users' notion of document relatedness. To the best of our knowledge, there have been no empirical studies of how users' characterization of related documents, expressed in natural language (NL) statements, maps onto the concept graphs and how the users' notion of relatedness compares with the metrics of KB-based algorithms. The former is particularly important to generate a NL explanation of relevance metrics and search results. Our research aims to fill that gap.

Starting with the users' explanations of how pairs of documents are related, we aim to devise methods that can identify concepts in a KB graph that express the same notion of document relatedness. Our algorithm first maps document terms onto KB concepts and then generates paths through the KB graph that originate from document concepts. For a pair of documents we can then identify all concept paths that intersect and therefore have a KB concept in common. Such KB concepts capture the semantic relatedness of document terms and the corresponding documents. We score and rank common concepts and use them to characterize the document relatedness.

That brings us to the final step of verifying how common document concepts relate to the users' explanations of document relatedness. We map the users' expressions onto the KB concepts and observe how high they appear in the ranked list of intersecting concepts. The higher they rank, the better the method captures the users'

CIKM '16, October 24–28, 2016, Indianapolis, IN, USA.

DOI: http://dx.doi.org/10.1145/2983323.2983913

notion of document relatedness. Findings of our experiments show that the algorithm that fully exploits KB structure by considering KB concept links performs significantly better than a method that relies upon concept statistics only.

2. RELATED WORK

Definition and implementation of document similarity is intrinsically related to the underlying IR approach and varies significantly with the adopted document representation, as can be seen from research in vector-space and probabilistic models [9, 8], latent semantic indexing [2], and KB-supported models [10, 6].

Use of KB-based document representations presents new opportunities to tackle two important retrieval challenges: (1) capture concepts that are common to two documents but not explicitly mentioned in the documents and (2) communicate to the user the rationale behind the document relatedness in a user-interpretable form through concepts linked to the document terms.

Recent work with KB-based document similarity[10, 6] focuses on entity linking to a specific KB (e.g., DBPedia or BabelNet), with the aim of producing a document similarity score. For example, Schuhmacher and Ponzetto [10] measure similarity in terms of a graph edit distance between respective document subgraphs using weighted concept edges. They do not explore in detail the conceptual links between documents and the relative importance of linked KB concepts within documents.

Ni et al. [6] extend the model from [10] by assigning different weights to linked concepts and allowing for inexact node matching by a means of comparing their *embedding vectors*. They derive concept weights from concept properties in the KB without taking into account term mentions in the document (e.g., term frequency). Since concepts are compared within an embedding space, the potential for generating user-understandable explanations of document relatedness is lost.

None of the above methods relate document similarity metrics to the users' notion of document relatedness. We argue that this is important in order to maximize the benefits of semantically rich representations that KBs offer. To this end, we explore ways to incorporate users' expressions of relatedness into the KB representations and similarity operators. We use BabelNet [5] to generate document concept graphs and identify shared concepts on the connecting paths that capture document relatedness. We analyze the set of shared concepts with regards to the concepts in the users' explanations of document relatedness to see to what degree our algorithms capture the users' notion of semantic similarity.

3. METHODOLOGY

In this section we describe the algorithm for recognizing and ranking conceptual links between the documents. Although in our experiments we use BabelNet as the KB and Babelfy [4] as the concept linker, the algorithm itself is KB and linker independent.

3.1 Concept Intersection Graph

The first step in creating a document concept graph is linking the terms mentioned in the document with concepts in the KB. Let d be an input document and $B = (V_B, E_B)$ the KB graph in which nodes denote the concepts contained in the KB and edges capture semantic relations between concepts. Let $V_d \subset V_B$ be the set of KB concepts to which the linker connects document terms.

In order to relate two documents, we consider paths in B that connect any concept linked in the first document with any concept in the second document. Due to the size and density of KBs, such exhaustive search for concept paths is computationally intractable. Thus, we reduce the search space by limiting the path length, starting with the set of document concepts V_d of linked entities and creating a *concept front*, i.e., a subgraph of B that contains all nodes that are up to n edges distant from the nodes in V_d.

Let $B_i = (V_{B_i}, E_{B_i})$ and $B_j = (V_{B_j}, E_{B_j})$ be the concept fronts of size n for documents d_i and d_j, respectively. We merge the two concept fronts into a single graph, $B_{ij} = (V_{B_i} \cup V_{B_j}, E_{B_i} \cup E_{B_j})$ and examine the *candidate concepts* from the intersection of the two concept fronts, $V_F = V_{B_i} \cap V_{B_j}$. Since we use breadth-first-search strategy to build the concept fronts, each candidate concept node, $v_f \in V_F$, must have at least one path to at least one concept linked from each of the input documents. By merging these two paths on the shared node $v_f \in V_F$, we obtain an *inter-document path* from the concept $v_i \in V_{d_i}$ to the concept $v_j \in V_{d_j}$. After collecting all such document connecting paths, we proceed by assigning a *relatedness score* to each of them. In fact, we consider several characteristic of documents and concepts to define the score.

3.2 Scoring Paths and Concepts

Our objective is to characterize the concepts $v_f \in V_F$ found in the intersection of the two concept fronts and, as an intermediate step, we score directional inter-document paths through the concepts, $p = \{(v_1, v_2), \ldots, (v_k, v_{k+1}), \ldots (v_{n-1}, v_n)\}$. The starting node of the path $v_1 \in V_i$ denotes the concept linked from the first input document d_i, the ending node $v_n \in V_j$ denotes the concept linked from the second document d_j, and an intersecting node $v_k \in V_F$ is one of the candidate concepts that reflects the relatedness of the concepts v_1 and v_n. For each such path p, we compute several features with which we aim to capture the relevance of the path for the explanation of commonalities between the documents.

Salience of linked entities (SAL). SAL aims to capture prominence of KB concepts within documents. For the sake of simplicity, we adopt document term frequency as a measure of SAL. Thus, for each directional path from the document d_i to the document d_j, we compute the SAL score for the concept $v_1 \in V_{d_i}$ at the start and the concept $v_n \in V_{d_j}$ at the end of the document connecting path.

Linking confidence (LC). LC is a measure of confidence that a term mentioned in a document is linked to the correct concept in the KB. For each document connecting path $\{(v_1, v_2), \ldots, (v_{n-1}, v_n)\}$, we compute LC scores for nodes v_1 and v_n, with the aim of reducing the contribution of paths that connect document terms that are more likely to be incorrectly linked to the KB concepts.

Specificity of path relations (SR). We adopt the assumption from [10] that more specific relations between concepts are more informative and thus more effective in supporting IR tasks. Let $r(v_k, v_{k+1})$ be the semantic relation of an edge (v_k, v_{k+1}) and R the set of all KB relations. We compute the relation relevance score of path p as the average of information contents of individual relations along p:

$$spec(p) = \frac{1}{|p|} \sum_{(v_k, v_{k+1}) \in p} -\log \frac{freq(r(v_k, v_{k+1})) + 1}{|R| + \sum_{r' \in R} freq(r')}$$

where $freq(r)$ is the number of occurrences of relation r in the KB.

Specificity of path concepts (SC). For each node v along the path, we compute the specificity of the corresponding concept c in terms of the relative frequency of c in a large corpus D:

$$ic(c, D) = -\log \frac{freq(c, D) + 1}{|D| + \sum_{c' \in D} freq(c', D)}$$

This is motivated by the assumption that terms that occur rarely in the corpus better describe the document and are better indicators of document relatedness. We used the Google Book Ngrams [3] as a

large corpus D to estimate term specificity. The feature score for the whole path is the average of SC of individual nodes.

Path symmetry (SYM). Another assumption we make is that an explanatory path in the KB is more relevant for a given pair of documents if it is more symmetric with respect to some candidate concept node $v_f \in V_F$. A symmetric explanatory path means that the same set of semantic relations are required to traverse (1) from a concept linked from the first document (denoted by v_1) to the candidate concept v_f and (2) from a concept linked from the second document (denoted by v_n) to the candidate concept v_f. Let $p_1 = \{(v_1, v_2), \ldots, (v_{k-1}, v_k)\}$ and $p_2 = \{(v_k, v_{k+1}), \ldots, (v_{n-1}, v_n)\}$ (with $v_k \in V_F$) be the two subpaths of the inter-document path p and let $r(p)$ denote the set of KB relations on the path p. We then quantify the symmetry of the path as follows:

$$sym(p) = \frac{|r(p_1) \cap r(p_2)|}{|p|}$$

Ranking common concepts. Because the scores of above features are on different scales, we standardize the scores for each feature (z-score) before combining them into a single score for a path. We compute the final relatedness score of the path p by averaging all the feature scores assigned to that path. Finally, we compute the score for each node v_f in the front intersection V_F by summing the scores of all the paths that contain v_f. The algorithm outputs the ranked list of the KB concepts in V_F. In cases when the same concept has been found in both documents (i.e., $v_1 = v_n$), the connecting path p degenerates to an empty sequence, and the *path symmetry* and *specificity of path relations* are assigned the maximum z-score.

Complexity. The time complexity of the algorithm, stemming from the breadth-first search for the front concepts V_F, is $O(|V_d|b^{n+1})$, where b is the branching factor of nodes in the KB and n the distance in edges from the starting node. The branching factor is not constant. We estimate $b = 10$ by averaging out-degrees of all nodes traversed in BabelNet. In our experimental setting, we used $n = 5$ and the average number of concepts in the intersection was $\hat{V_F} = 10000$.

4. EVALUATION

As noted in the introduction, research efforts in KB-based representation of documents have primarily focussed on characterizing document relatedness through computing a semantic similarity score. Evaluations were thus conducted by measuring correlation between user- and method-assigned scores. In contrast, we score and rank concepts from a KB that connect paths originating from compared documents, thus reflecting the document relatedness. In our evaluations, we observe how they relate to the concepts expressed by the users when articulating their views about document relatedness.

4.1 Dataset Preparation

The dataset used in our experiments consists of pairs of news articles for which connecting concepts need to be identified by the users and the algorithms, and compared. In order to ensure that there are conceptual links between documents, we guided the news pairing process. However, as the procedure below shows, we have not introduced any bias in favor of our algorithm:

1. We asked three annotators to come up with short lists of concepts for three types of notions – general (e.g., *"shark attack"*), named entities (e.g., *"Elon Musk"*), and events (e.g., *"FA Cup final"*);
2. We used the provided concepts as queries for online web search engines and obtained a collection of 25 news stories;
3. We computed cosine similarity between term vectors of the retrieved news articles. We then randomly paired each news story d_i with two other stories d_j and d_k, so that the probability of randomly selecting d_j and d_k was proportional to their cosine similarity with d_i. In other words, documents with larger cosine similarity were more likely to be paired.

Table 1: Annotation example

Story #1. *"A pet shop owner has survived an attack by a deadly, 20ft python after police wrestled the snake back into its cage. Terry Wilkins was cleaning the reptile's cage at his store in Kentucky, on Monday when it bit him, wrapped itself round his head and started crushing his neck. The snake had covered Mr Wilkins' face and was moving in for the kill when two nearby police officers intervened and saved his life."*
Story #2 *"A 9-year-old Northern California boy was mauled to death by his sister's pit bulls while she was at her job Sunday morning, police said. Investigators believe the three pit bulls Alexandria Griffin-Heady calls her "wolfpack" attacked Tyler Griffin-Huston during his 24-year-old sister's approved weekend custody of him at her Linda residence, KHTK-TV reported."*
Raw annotations: Both stories recount animal attacks on humans; In both cases, the victim is male; Both incidents happened in the US; In both cases, the police were involved;
BabelNet codifications: animal (bn:00004222n), attack/assault (bn:00006467n), human (bn:00044576n); male/man (bn:00001533n); United States (bn:00003341n); police (bn:00022026n);

Following the above steps, we obtained a dataset comprising 50 pairs of news and instructed seven annotators to (1) indicate how two texts are related, i.e., what common aspects link them, (2) describe the common aspects in their own words, and (3) rank the aspects in the decreasing order of importance. We intentionally left the annotation task underspecified in order to elicit all types of connections between two articles that the users may find and articulate. We made sure that each article pair was processed by two annotators so that we can determine the annotators' agreement.

To compare the relatedness of news stories based on users' assessment and based on our algorithm, we manually mapped the annotators' NL descriptions to BabelNet concepts. This step was carried out by a single annotator to ensure that the same users' notions were consistently linked to the same BabelNet concepts. The annotation is exemplified in Table 1. As expected, not all aspects identified by the annotators could be directly mapped onto BabelNet concepts, including statements such as *"Both news stories are written in a negative tone"* or *"Chances of events happening are relatively small"*. We labelled such explanations as *not-codifiable* and penalized our approach in the experimental evaluation as not being able to generate KB based representation of users' explanations. We measured the inter-annotator agreement in terms of the (rank-agnostic) F_1 measure and observed a very low score of 11.4%. This confirms the highly subjective nature of the task. Thus we chose not to resolve annotators' disagreements but, rather, evaluate the methods against each annotator's input and then compute the average performance for each document pair.

4.2 Experiments Design and Findings

We designed our experiments to compare (1) KB concepts used by the annotator to describe pairwise document relatedness against (2) KB concepts identified and ranked by several algorithms. We observe where the annotators' concepts can be found in the ranked lists of concepts returned by the algorithms.

Evaluation metrics. We initially applied the mean average precision (MAP) by averaging precision scores at the rank positions where the annotators' concepts are observed. However, (1) MAP scores are not informative when there is a large imbalance between the size of the ranked list and the number of relevant items, as in

Table 2: Algorithm performance

Algorithm	MAP (%)	MWAR
N-gram overlap	8.65	4012.3
Concept overlap	10.46	3789.4
KB ranker	**10.94**	**3020.4**

our case where algorithms return thousands of ranked KB concepts while annotators provide only a handful, and (2) MAP metrics cannot take into account the concept ranking provided by the annotators. For these reasons we introduced the *mean weighted average rank* (MWAR) metric. For a given document pair, let G be the ranked list of concepts provided by an annotator and let A be the list of concepts produced by an algorithm (typically $|A| \gg |G|$). With $a(G_i)$ being the rank in A of the i-th concept from the annotator's list, the *weighted average rank* (WAR) is computed as follows:

$$WAR(G, A) = \frac{\sum_{i=1}^{|G|} (|G| - i + 1) \cdot a(G_i)}{\sum_{i=1}^{|G|} i}$$

The MWAR is simply the mean of WAR scores over all document pairs (lower is better). The annotators' concepts that are not present in the concept lists produced by the algorithm are assigned rank $|A| + 1$. The two baselines that we use produce significantly shorter rankings than our KB-based algorithm. To allow for fair comparison, we also assign rank $|A| + 1$ to gold concepts when they are missing from the baseline ranking.

Parameter optimization. We split the dataset of 50 article pairs into the development set (10 pairs) and test set (40 pairs). We use the development set to determine the optimal value for the path lengths, i.e., front distance; we set it to $n = 4$ (cf. Section 3.1).

Baselines. We compare the performance of our KB-based algorithm with two baseline algorithms. The first algorithm is the *n-gram overlap*, which identifies term n-grams that occur in both documents and ranks them in decreasing order of frequencies. The second baseline, *concept overlap*, considers KB concepts that literally appear in the document. It ranks the common concepts based on their frequency in the two texts. These baselines are selected to contrast our approach with one algorithm that does not utilize KB in any way and another that considers only KB concept nodes that appear in the texts while it does not utilize KB links and relatedness to other concepts.

Results. Table 2 shows the performance of our KB-based algorithm (*KB ranker*) and the two baselines. The KB-based algorithm outperforms both baselines, for MWAR significantly ($p < 0.05$, Student's t-test). This justifies the use of KB to explain document relatedness by (1) seeking conceptual links as opposed to simple term overlap, and (2) identifying concept paths and common frontal concepts.

5. CONCLUSION

One of key benefits of the KB based document representations is the availability of explicit semantic links among concepts that can be used to describe the relation between documents in a user-interpretable form. Similarly, a user could describe, in natural language, desired properties and relatedness of relevant documents. Such a specification would be then expressed in terms of KB concepts and links and used to identify documents that fit the users' criteria. In both instances, it is critical to compare users' expressions with those KB concepts that algorithms use to characterize semantic relevance of documents.

We presented a method and experiment results of comparing users' expressions of document relatedness with several algorithms designed to work with document concept graphs to measure semantic similarity. Our initial results show that algorithms that exploit full KB structure lead to a better matching of users' expressions than those relying upon simple lexical overlap or concept statistics only.

Our further research will include systematic evaluation and optimization of algorithm components and improvements of concept rankings. That will include larger datasets and more intuitive and versatile annotation schemes.

6. ACKNOWLEDGMENTS

This work has been funded by the Unity Through Knowledge Fund of the Croatian Science Foundation, under the grant 19/15: "EVEnt Retrieval Based on semantically Enriched Structures for Interactive user Tasks (EVERBEST)". The research has been carried out within the activities of the Centre of Research Excellence for Data Science and Cooperative Systems supported by the Ministry of Science, Education and Sports of the Republic of Croatia. We thank the annotators for their efforts and the reviewers for their comments.

7. REFERENCES

[1] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[2] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391, 1990.

[3] J.-B. Michel, Y. K. Shen, A. P. Aiden, A. Veres, M. K. Gray, J. P. Pickett, D. Hoiberg, D. Clancy, P. Norvig, J. Orwant, et al. Quantitative analysis of culture using millions of digitized books. *Science*, 331(6014):176–182, 2011.

[4] A. Moro, A. Raganato, and R. Navigli. Entity linking meets word sense disambiguation: a unified approach. *Transactions of the ACL*, 2:231–244, 2014.

[5] R. Navigli and S. P. Ponzetto. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250, 2012.

[6] Y. Ni, Q. K. Xu, F. Cao, Y. Mass, D. Sheinwald, H. J. Zhu, and S. S. Cao. Semantic documents relatedness using concept graph representation. In *Proceedings of ACM WSDM '16*, pages 635–644. ACM, 2016.

[7] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proceedings of ACM SIGIR '98*, pages 275–281. ACM, 1998.

[8] S. Robertson and H. Zaragoza. *The probabilistic relevance framework: BM25 and beyond*. Now Publishers Inc, 2009.

[9] G. Salton, A. Wong, and C.-S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[10] M. Schuhmacher and S. P. Ponzetto. Knowledge-based graph document modeling. In *Proceedings of the ACM WSDM '14*, pages 543–552. ACM, 2014.