A Discourse Search Engine based on Rhetorical Structure Theory

Pascal Kuyten, Danushka Bollegala, Bernd Hollerit, Helmut Prendinger, and Kiyoharu Aizawa

National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo, 101-8430, Japan.
The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-8656, Japan.
The University of Liverpool, Liverpool, L693BX, United Kingdom.
pascal@kuyten.com, hollerit@gmail.com, helmut@nii.ac.jp

Abstract. Representing a document as a bag-of-words and using keywords to retrieve relevant documents have seen a great success in large scale information retrieval systems such as Web search engines. Bag-of-words representation is computationally efficient and with proper term weighting and document ranking methods can perform surprisingly well for a simple document representation method. However, such a representation ignores the rich discourse structure in a document, which could provide useful clues when determining the relevancy of a document to a given user query. We develop the first-ever *Discourse Search Engine* (DSE) that exploits the discourse structure in documents to overcome the limitations associated with the bag-of-words document representations in information retrieval. We use Rhetorical Structure Theory (RST) to represent a document as a discourse tree connecting numerous elementary discourse units (EDUs) via discourse relations. Given a query, our discourse search engine can retrieve not only relevant documents to the query, but also individual statements from those relevant documents that describe some discourse relations to the query. We propose several ranking scores that consider the discourse structure in the documents to measure the relevance of a pair of EDUs to a query. Moreover, we combine those individual relevance scores using a random decision forest (RDF) model to create a single relevance score. Despite the numerous challenges of constructing a rich document representation using the discourse relations in a document, our experimental results show that it improves the F-score in an information retrieval task. We publicly release our manually annotated test collection to expedite future research in discourse-based information retrieval.

1 Introduction

In a typical bag-of-words (BOW) approach to document representation, first a document is tokenized into a set of tokens (often unigrams or bigrams), next a pre-defined set of stop words is removed from the tokens, and finally the remainder of the tokens are used as index entries to build an inverted index. When a user of a search engine enters keywords (often one or two words) describing her information need, those keywords are matched against the inverted index, and matching documents are returned to the user. If the number of search results is large as in a typical web search engine, accurate ranking of search results, considering the relevance of a document to the user query,

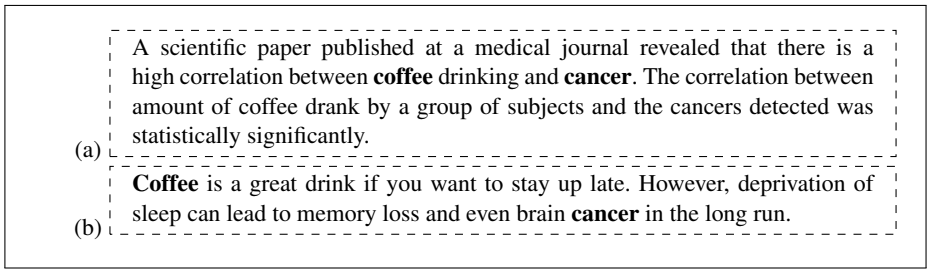

Fig. 1. Two documents mentioning the two terms *coffee* and *cancer*. Document (a) describes an EVIDENCE discourse relation between the two sentences, whereas in document (b), there is a CONTRAST discourse relation between the two sentences. For a user who searches for evidence that supports the claim *coffee causes cancer*, document (a) is more relevant than (b).

becomes important. Although the BOW representation is attractive for its robustness and efficiency, which are indeed vital factors when considering the scale and the quality of the documents found on the Web, a natural question is *whether IR can benefit from linguistically rich document representations beyond the BOW approach?*. We address this question by proposing and evaluating a document representation method based on the discourse relational structure in a document.

Despite its simplicity and popularity, the BOW representation of documents in IR systems ignores the rich discourse structure embedded in the documents, which can provide useful clues when determining the relevance of a document to a user query. For example, consider the two documents shown in Fig. 1. A user who is interested in evidence that supports the claim *coffee causes cancer* would benefit from the document (a) than the document (b). However, the BOW representations for each document contain both words *coffee* and *cancer*. Consequently, a search engine that indexes documents represented as bags-of-words will be unable to differentiate the subtle differences of the relevancies of the documents to user queries.

Discourse theories such as the Rhetorical Structure Theory (RST) [14] represent a document using a set of discourse units, connected via a pre-defined set of discourse relations (e.g. ELABORATION, CONTRAST, and JUSTIFICATION). For example in Fig. 1, the two sentences in document (a) and (b) are connected respectively through ELABORATION and CONTRAST relations. An IR system that utilizes the discourse structure of a document will be able to rank document (a) higher than (b) for the query *coffee causes cancer*, thereby improving the user satisfaction. Discourse information has shown to improve performance in numerous related tasks in natural language processing (NLP) such and text summarization [13].

Despite the benefits to an IR system from a discourse-based representation of documents, building discourse-aware IR systems is a challenging task due to several reasons. First, accurately identifying the discourse relations in natural language texts is difficult. Discourse markers such as *however, but, contrastingly, therefore,* etc. can be ambiguous with respect to the discourse relations they express [7]. It is inadequate to classify discourse relations purely based on discourse markers, and discourse parsers that use more advanced NLP methods are required [5,6,9,10,12,19,22]. Second, not all types of

natural language texts are amenable to discourse parsers. For example, unlike newspaper articles, scientific publications, or Wikipedia articles that are logically structured and proofread, most texts found on the Web do not possess a well-organized discourse structure. Third, relevance measures that capture the underlying discourse structure of documents are lacking. It is non-obvious as to which discourse relations are useful for IR. Fourth, there does not exist any benchmark test collections that are annotated with discourse information for IR. It is difficult to empirically evaluate the pros and cons of discourse-motivated IR systems at larger scales without having access to discourse-annotated test collections.

We propose *Discourse Search*, a novel search paradigm that goes beyond the simple BOW representations of documents and captures the rich discourse structure present in the documents. First, we segment each document into Elementary Discourse Units (EDUs). An EDU is defined as a single unit of discourse and can be either a clause, a single sentence, or a set of sentences. Next, we identify EDU pairs that have some discourse relations according to RST. Discourse relations proposed in RST are directional relations and distinguish the main and the subordinate EDUs involved, referred to as respectively the *nucleus* and the *satellite*. Finally, all EDUs are arranged into a single binary tree structure covering the entire document. We index each sub-tree consisting of a pair of EDUs and a discourse relation. During retrieval time, we match a user query against this index and return pairs of EDUs as search results to the user. In particular, our discourse search engine goes beyond document-level IR and can retrieve the exact statements from the relevant documents. This is particularly useful when a single document expresses various opinions about a particular topic.

Our contributions in this paper can be summarized as follows.

- We develop a *Discourse Search Engine* (DSE) that considers the discourse structure present in documents to measure the relevance to a given user query. To our knowledge, ours is the first-ever IR system that uses RST to build a DSE.
- We propose three discourse proximity scores to measure the relevance of a pair of EDUs to a user query, considering the discourse structure in a document. Moreover, we learn the optimal combination of those three scores using random decision forests.
- We create a test collection annotated with discourse information to evaluate discourse-based IR systems. Specifically, for each test query, the created test collection contains a ranked list of EDU pairs indicating their relevance to the query. Considering the immense impact that test collections such as TREC benchmarks has had upon the progress of the research in IR, we publicly release the created test collection to expedite the future research in discourse-based IR.

2 Related Work

The use of discourse analysis as a tool for studying the interaction between a user and an IR system dates back to early 80's work of Brooks and Belkin. [3]. The task of retrieving information related to a particular information need is seldom a one-step process, and requires multiple interactions with the IR system. By analyzing this dialogue between a user and an IR system, we can improve the relevance of the retrieved search results.

For example, by using search session data, it is possible to accurately predict the user intent [16]. Our work in this paper is fundamentally different from this line of prior work, because we are analyzing the discourse structure in the *documents* instead in the *dialgoues* between a user and a search engine.

Wang et al. [21] classified queries based on their discourse types and proposed a graph-based re-ranking method. In particular, they considered queries that describe an information need related to the advantages and disadvantages of a particular decision (e.g. *What are the advantages and disadvantages of same-sex schools?*). The relevance between a query and a document is measured using a series of proximity-based measures. However, unlike our work, they do not consider the discourse structure present in the documents. Moreover, our DSE is not limited to a particular type of discourse queries, and supports a wide-range of queries.

Using semantic relations that exist between entities in a document to improve IR has received wide-attention in the NLP community. For example, in Latent Relational Search [4], given the two entities *YouTube* and *Google* as the query, the objective is to retrieve other pairs of entities between which the same semantic relations exist. Here, the semantic relation ACQUISITION holds between *YouTube* and *Google*. Therefore, other such pairs of entities where one of the entities is acquired by the other such as, *Powerset* and *Microsoft* are considered as relevant search results. Latent relational search can be classified as an instance of analogical search, where the focus is on the semantic relations between the entities and not the entities themselves. Latent relational search engines represent the semantic relations between two entities using a vector of lexical pattern frequencies, and measure the relational similarity between two pairs of entities by the cosine similarity between the corresponding lexical pattern frequency vectors. Interestingly, this approach can be extended to cross-language relational search as well.

Miyao et al. [15] developed a search engine for Bio-medical IR by extracting the semantic relations that are common in the Bio-medical domain such as, the interaction between proteins, or side-effects of a drug. First, they apply a term extraction method to detect Bio-medical terms in the documents, and extract numerous features from an Head-driven Phrase Structure Grammar (HPSG) parse tree of a sentence. A bio-medical relation classifier is trained using the extracted features. Although semantic relations are useful as an alternative to the BOW representation, it is complementary to the discourse structure that we exploit in our DSE. Indeed, an interesting future research direction would be to combine both semantic relations and discourse relations to further improve the performance of IR systems.

3 Rhetorical Structure Theory

We briefly review Rhetorical Structure Theory (RST) [14] that defines the discourse structure that we use in our document representation. In RST, documents are segmented into non-overlapping elementary discourse units (EDUs). EDUs are related by a discourse relation, where the head EDU (*nucleus*), has a relation with the subordinate EDU (*satellite*). EDUs are arranged into a binary tree to create a *discourse tree* for a document. Directed edges of a discourse tree point from a satellite to a nucleus and are labeled with a discourse relation. In RST, nuclei and satellites may consist of sin-

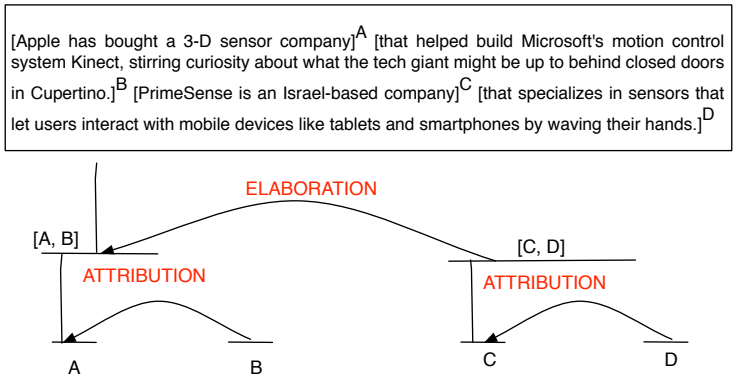

Fig. 2. A discourse tree covering four EDUs.

gle or multiple EDUs in the latter case, the individual EDUs are related by a path of discourse relations. An example of a discourse tree is shown in Fig. 2, covering four EDUs, where ATTRIBUTION relations exist between the two EDUs in each sentence, and an ELABORATION relation holds between the two sentences.

Discourse trees can be automatically generated using discourse parsers such as SPADE [18], or HILDA [8]. SPADE produces sentence level discourse structures, whereas a complete discourse tree covering all the sentences in a document can be generated using HILDA. Because our goal is to represent entire documents considering their discourse structures, we use HILDA as our preferred discourse parser. HILDA builds a single discourse tree by segmenting the document into EDUs using a Hidden Markov Model (HMM). Next, from these EDUs, a single discourse tree is built by discourse relation classification using Support Vector Machines (SVMs) [5]. HILDA's classifies 18 discourse relation types such as, ELABORATION, ATTRIBUTION, and CONTRAST.

4 Discourse Search Engine

Let us denote the discourse tree of a document d by $\mathcal{T}(d) = \{(\delta_n, \delta_s, r(\delta_n, \delta_s))\}$, where a discourse relation $r(\delta_n, \delta_s)$ holds between a nucleus δ_n and a satellite δ_s in the discourse tree. To simplify the notation, we will write r in place of $r(\delta_n, \delta_s)$, when it is clear from the context as to r holds between which two EDUs. For example, the document shown in Fig. 2 is represented by the set consisting of the three elements: $(A, B, \text{ATTRIBUTION})$, $(C, D, \text{ATTRIBUTION})$, and $([A, B], [C, D], \text{ELABORATION})$. Here, $[A, B]$ denotes the parent vertex of EDUs A and B. Given d, a discourse parser can be used to generate $\mathcal{T}(d)$.

Likewise, we define a discourse query Q as a three-valued tuple $(q_n, q_s, r(q_n, q_s))$, where a discourse relation $r(q_n, q_s)$ holds between the nucleus q_n and the satellite q_s of Q. For example, the query *coffee causes cancer* is mapped to the tuple (*coffee*, *cancer*, EVIDENCE). Information needs of a user can be mapped into a discourse query by several methods. Given a natural language input such as *coffee causes cancer*, a discourse parser can be used to generate a discourse query. Alternatively, we could train

a sequence labeller such as a conditional random field [11], to extract the two EDUs and the discourse relation between them. A more manual approach would be to provide a search front end where a user can enter the nucleus, satellite and select a discourse relation from a drop-down list. The DSE we propose can be easily incorporated with all of those approaches.

We model the relevancy of a discourse query Q to a document d as a function $f(Q, d)$, which is the summation of the product between a **discourse relation selector**, $\phi(Q, \delta_n, \delta_s, r) \in [0, 1]$, and a **discourse proximity score**, $\psi(Q, \delta_n, \delta_s, \mathcal{T}(d)) \in [0, 1]$, over all EDU pairs in the discourse tree $\mathcal{T}(d)$ as follows

$$f(Q, d) = \sum_{(\delta_n, \delta_s, r) \in \mathcal{T}(d)} \phi(Q, \delta_n, \delta_s, r)\psi(Q, \delta_n, \delta_d, \mathcal{T}(d)). \quad (1)$$

For the document shown in Fig. 2, all possible combinations between δ_n and δ_s are listed in Table 2. Next, we will discuss each of those factors in detail.

A DSE must consider the agreement between the discourse relation $r(q_n, q_s)$ in the query, and the relation $r(\delta_n, \delta_s)$ between two EDUs δ_n, δ_s in the document. Moreover, not all words are equally significant when considering the relevance between a query and a document. For example, frequent non-content words are removed from the queries using a pre-defined stop-words list by most search engines, and term-weighting scores such as tfidf, or BM25 are used to detect salient matches. We propose discourse relation selector, $\phi(Q, \delta_n, \delta_s, r)$, as a function that captures those two requirements. It is defined as follows:

$$\phi(Q, \delta_n, \delta_s, r) = s(q_n, \delta_n)s(q_s, \delta_s)\mathrm{I}[r(q_n, q_s) \in l(\delta_n, \delta_s, \mathcal{T}(d))]. \quad (2)$$

Here, $s(w, \delta)$ is a salience score such as, tfidf or BM25 indicating the salience of a word w in an EDU δ in the discourse tree $\mathcal{T}(d)$, and $\mathrm{I}[r(q_n, q_s) \in l(\delta_n, \delta_s, \mathcal{T}(d))]$ is the indicator function which returns 1 if the discourse relation $r(q_n, q_s)$ between q_n and q_s appears in the discourse path $l(\delta_n, \delta_s, \mathcal{T}(d))$ between EDUs δ_n, δ_s in the document, and 0 otherwise. For example, the discourse path between EDUs A and C shown in Fig. 2 is $A \rightarrow [A, B] \xrightarrow{\text{ELABORATION}} [C, D] \leftarrow C$. It contains the ELABORATION discourse relation between A and C. In our experiments, we used tfidf as the salience score $s(w, \delta)$, and consider the occurrences of query words in the EDUs extracted from the documents. Because a single document can contain multiple topics, we found it is more accurate to compute tfidfs over EDUs than entire documents. Using the Porter's stemming algorithm[1] we perform stemming on the words in the documents before computing tfidf scores.

The location of the words used in the query in their appearance in the document is an important feature that influences the relevance of the document to the query. For example, if all the words used in the query appear within close proximity in the document, the higher is the relevance [1]. We adopt this intuition to discourse trees by proposing three types of discourse proximity scores for $\psi(Q, \delta_n, \delta_s)$ as we describe next.

The first of the three discourse proximity scores we propose, **segment proximity**, $\psi_{seg}(Q, \delta_n, \delta_s, \mathcal{T}(d))$, measures the distance between two EDUs δ_n, δ_s as the number

[1] `http://tartarus.org/martin/PorterStemmer/`

of discourse segments (EDUs) that appear in between δ_n and δ_s in the document. The segment proximity is given by,

$$\psi_{seg}(Q, \delta_n, \delta_s, \mathcal{T}(d)) = 1 - \frac{|t(\delta_n, \mathcal{T}(d)) - t(\delta_s, \mathcal{T}(d))| - 1}{\mathbf{E}(d) - 2}. \quad (3)$$

Here, $t(\delta, \mathcal{T}(d))$ indicates the segment number (starting with 1 and counted from the beginning of the document) of the EDU δ in the discourse tree $\mathcal{T}(d)$, and $\mathbf{E}(d)$ denotes the total number of EDUs in the document. $\psi_{seg}(Q, \delta_n, \delta_s)$ is normalized by dividing from $\mathbf{E}(d)$ to remove any biases due to differences in document lengths. If two EDUs δ_n, δ_s are closer to each other in the document, the higher their segment proximity will be. For the example shown in Fig. 2, the four EDUs appear in the order $t(A, \mathcal{T}(d)) = 1$, $t(B, \mathcal{T}(d)) = 2$, $t(C, \mathcal{T}(d)) = 3$, and $t(D, \mathcal{T}(d)) = 4$ in the document text. Therefore, for example, $\psi_{seg}(Q, A, C) = 1 - ((|1 - 3| - 1)/(4 - 2)) = 1/2$.

Two EDUs that appear in distant locations on the surface text of a document, might have a direct discourse relation between them. Such EDUs appear close together on the discourse tree, despite being located far apart on the surface text of the document. The segment proximity would assign a low score for such related EDUs because it only considers the surface text and ignores the discourse tree structure. We propose **path proximity**, $\psi_{path}(Q, \delta_n, \delta_s, \mathcal{T}(d))$, as a measure that computes the closeness between two EDUs δ_n, δ_s over the discourse tree $\mathcal{T}(d)$ by the length of the discourse path $l(\delta_n, \delta_s, \mathcal{T}(d))$ that connects δ_n to δ_s. Specifically, path proximity is given by,

$$\psi_{path}(Q, \delta_n, \delta_s, \mathcal{T}(d)) = 1 - \frac{|l(\delta_n, \delta_s, \mathcal{T}(d)| - 1}{\log_2 \mathbf{E}(d)}. \quad (4)$$

Here, $|l(\delta_n, \delta_s, \mathcal{T}(d)|$ denotes the length of the discourse path connecting δ_n to δ_s, and is measured by the number of discourse relations (ignoring the directions) along the discourse path. For example, the discourse path between EDUs A and C shown in Fig. 2, $A \rightarrow [A, B] \xrightarrow{\text{ELABORATION}} [C, D] \leftarrow C$, contains one discourse relation, ELABORATION, resulting in a length of 1. The $\log_2 \mathbf{E}(d)$ term in the denominator is the diameter of the discourse tree (i.e. maximum distance between any two vertices), and is derived using the property that discourse trees are binary trees. For example, the path proximity $\psi_{path}(Q, A, C, \mathcal{T}(d))$ between A and C is computed as,

$$\psi_{path}(Q, A, C, \mathcal{T}(d)) = 1 - \frac{1 - 1}{\log_2 4} = 1.$$

The first occurrence of an entity in a document often contains its definition. For example, in news text summarization, the first sentence baseline where the first sentence (also known as the lead sentence) is used as the summary of the document [13]. We translate this heuristic to measure the relevance of a query to a discourse tree by considering the shortest segment distance from the first EDU to the two discourse units δ_n and δ_s that contain respectively q_n and q_s. We refer to this relevance score as the **Lead EDU Proximity** score, which is given by,

$$\psi_{lead}(Q, \delta_n, \delta_s, \mathcal{T}(d)) = 1 - \frac{\min\left(t(\delta_n, \mathcal{T}(d)), t(\delta_s, \mathcal{T}(d))\right) - 1}{\mathbf{E}(d) - 2}. \quad (5)$$

Similar to the segment proximity, we normalize the lead EDU proximity by dividing from the number of EDUs in the discourse tree to remove any bias due to the differences in document lengths. As an example, we compute the lead EDU proximity, $\psi_{lead}(Q, A, C, \mathcal{T}(d))$, between the two EDUs A and C in Fig. 2 as,

$$\psi_{lead}(Q, A, C, \mathcal{T}(d)) = 1 - \frac{\min(t(A, \mathcal{T}(d)), t(C, \mathcal{T}(d)) - 1}{4 - 2} = 1 - \frac{\min(1, 3) - 1}{2} = 0.$$

Recall that the overall relevance of a query Q to a document d is given by Equation 1 as the sum over the product of discourse relation selector, $\phi(Q, \delta_n, \delta_s, r)$, and each one of the three discourse proximity scores, $\psi(Q, \delta_n, \delta_s, \mathcal{T}(d))$. If there are no matching discourse relations between the query and a pair of discourse units selected from the document, then $f(Q, d)$ will be zero. We can use this fact to speed up the computation of $f(Q, d)$ in Equation 1 by not computing the discourse proximity scores for EDU pairs δ_n, δ_s for which $\phi(Q, \delta_n, \delta_s, r)$ is zero.

4.1 Combining Different Discourse Proximity Scores

Although we proposed three different discourse proximity scores for computing the relevance between a discourse query and a document it is not obvious as to the optimal combination of those discourse proximity scores that gives the best relevancy model. We model the problem of learning the optimal combination of discourse proximity scores as a learning-to-rank problem. Specifically, using a manually labeled dataset that lists a set of relevant documents for a discourse query, we follow a pairwise rank learning approach and train a binary classifier to detect relevant query-document pairs (positive class) from the irrelevant ones (negative class). Each query-document pair (Q, d) is represented by a three-valued feature vector using the relevance scores $f(Q, d)$ computed using each discourse proximity score in turn. Next, a Random Decision Forest (RDF) [2] is trained using the ALGIB[2] tool. The posterior probability, $p(+1|(Q, d))$, indicating the degree of relevance of Q to d is used as the combined relevancy score for the purpose of ranking documents retrieved for a discourse query[3]. All parameters of the RDF classifier are set to their default values as specified in ALGLIB.

4.2 Indexing and Query Processing

To efficiently process discourse queries, we create two inverted indexes: (1) an inverted index between all distinct n-grams in EDUs and the EDU IDs (similar to document IDs (urls) in traditional IR systems, we assign each EDU a unique ID) of the EDUs in which those n-grams occur, (2) an inverted index between nuclei EDU IDs and their corresponding satellite EDU IDs paired with the corresponding discourse relations. For the document shown in Fig. 2, an excerpt of the first inverted index is shown in Table 1, whereas Table 2 shows the corresponding second inverted index. Given a user query Q, we match the terms in q_n and q_s against the first index to find the matching EDUs.

[2] `http://www.alglib.net/dataanalysis/decisionforest.php#header3`

[3] Similarly, in a multi-class classifier, the posterior probability for the most probable class can be used as the ranking score.

Table 1. Excerpt of the inverted index between n-grams and EDU IDs for the document in Fig. 2.

Term	EDU ID
Apple	A
company	A, C
PrimeSense	C

Table 2. Inverted index between nucleus EDU IDs and their corresponding satellite EDU IDs with discourse relations.

Nucleus EDU ID	(Satellite EDU ID, discourse relations)
A	(B, ATTRIBUTION), (C, ATTRIBUTION, ELABORATION), (D, ATTRIBUTION, ELABORATION)
B	(C, ELABORATION, ATTRIBUTION), (D, ATTRIBUTION, ELABORATION)
C	(D, ATTRIBUTION)

Next, we use the second index to compute the discourse proximity scores. Finally, the set of EDUs that matches with the user query is ranked according to the relevance score $f(Q, d)$, computed using Equation 1 and returned to the user.

5 Evaluation

Evaluating an information retrieval system is a complex task involving numerous aspects such as, efficiency, accuracy, latency (indexing vs. query processing), scalability, and user satisfaction. Compared to keyword-based IR systems that have established evaluation measures and large test collections, discourse search is still in its early stages. To our knowledge, there does not exist an IR system that uses a document representation based on discourse relations, nor there exist benchmark test collections for discourse search. Therefore, an important contribution of our work is to create a test collection for evaluating discourse search engines for their accuracy. Section 5.1 describes the test collection we created for this purpose.

5.1 Dataset

We selected 10 online news articles covering news events related to major players in the IT industry such as (*Apple*, *Google*, *Microsoft*, *Facebook* and *Twitter*). We select major players in the IT industry to ensure our annotators, all graduate Computer Science students, would be familiar with the topic. Next, we generate a discourse tree, $\mathcal{T}(d)$, from each document d using the HILDA [8] discourse parser. Then, for each document we formulated a relevant query $Q(q_n, q_s, r(q_n, q_s))$ as (*main entity*, *related entity*, DISCOURSE RELATION). For example, a news article about *Microsoft* that introduces *Apple* as a competitor would result in the discourse query (*Microsoft*, *Apple*, ELABORATION). Finally, we extract multiple candidate EDU pairs (δ_n, δ_s) from each document that are connected by some discourse relation. HILDA segmented each document d into ca. 56 EDUs (min = 42, median = 57, max = 69), and ca. 6 candidate EDU pairs are selected from each $\mathcal{T}(d)$ (min = 4, median = 7, max = 10).

Table 3. Median values for discourse proximity scores

Grading	$\rho_{\mu-n}$	total no. of instances	ψ_{seg}	ψ_{path}	ψ_{lead}
$n = 2$ (irrelevant)	0	17	0.54	0.67	0.27
$n = 2$ (relevant)	1	40	0.09	0	0.35
$n = 4$ (irrelevant)	0	17	0.54	0.67	0.27
$n = 4$ (less relevant)	$\frac{1}{3}$	18	0.06	0.42	0.87
$n = 4$ (moderately relevant)	$\frac{2}{3}$	13	0.15	0	0.14
$n = 4$ (highly relevant)	1	9	0.25	0	0

Table 4. RDF performance for classifying EDU pairs for a query.

Features	$F(n = 2)$	$F(n = 4)$
$\psi_{seg}, \psi_{path}, \psi_{lead}$	0.75	0 .60
ψ_{seg}, ψ_{path}	**0.77**	0.54
ψ_{seg}, ψ_{lead}	0.75	0.58
ψ_{path}, ψ_{lead}	0.65	**0.61**
ψ_{seg}	0.74	0.32
ψ_{path}	0.68	0.26
ψ_{lead}	0.70	0.49

Six annotators individually read and rank 3 to 5 documents using a web interface during a 45 minute session. Documents were randomly distributed among the annotators, and we ensured each document was annotated by 3 to 5 annotators. The web interface first showed the instructions, then the annotators were asked to read a document. When an annotator clicked a button stating the document has been read, new instructions were presented. Next, a query and a set of candidate EDU pairs extracted from the document were presented. Annotators will mark an EDU pair as either relevant or irrelevant to a given query. Moreover, EDU pairs that are considered as relevant are further ordered according to their relevance to the query. Candidate EDU pairs were presented as complete sentences instead of segments by expanding the nucleus and the satellite to cover the entire sentences. For example, the EDU pair (A, C) in Fig. 2 is presented as (AB, CD) to the annotators. If both EDUs are in the same sentence only one sentence is presented. Our dataset is publicly available[4].

5.2 Results

Using the manually annotated dataset we created in Section 5.1, we evaluate the performance of the discourse proximity scores $\psi(Q, \delta_n, \delta_s, \mathcal{T}(d))$ we proposed in Section 4, by measuring the agreement between human annotations in the dataset and the relevance scores predicted by $f(Q, d)$. We denote the reciprocal of the rank given by the annotator a_i for the pair of EDUs (δ_n, δ_s), indicating its relevance to a query Q by $\pi(a_i, Q, \delta_n, \delta_s)$. The set of reciprocal ranks assigned by all annotators for a pair of EDUs (δ_n, δ_s) indicating its relevance to a query Q is denoted by $\rho(Q, \delta_n, \delta_s) = \{\forall_i | \pi(a_i, Q, \delta_n, \delta_s)\}$. We consider the majority vote, $\rho_\mu(Q, \delta_n, \delta_s)$, over the set of reciprocal ranks as the final relevance score of an EDU pair (δ_n, δ_s) to a query Q. Ties are

[4] `https://www.dropbox.com/s/7olgo2xk35yjkv2/collection.zip?dl=0`

resolved by selecting randomly between the majority reciprocal ranks. For example, if $\rho(Q, A, B) = \{\frac{1}{2}, \frac{1}{2}, 0\}$ then $\rho_\mu(Q, A, B) = \frac{1}{2}$. Considering the majority vote instead of the arithmetic mean has shown to improve the reliability when aggregating human ratings in annotation tasks [17]. For each query Q, we normalize the $\rho_\mu(Q, \delta_n, \delta_s)$ values for all candidate EDUs (δ_n, δ_s) retrieved for Q to the range $[0, 1]$ by fitting a uniform distribution. For example, given four EDU pairs, (A, B), (C, D), (E, F), and (G, H) retrieved for a query Q, if an annotator a labeled (A, B) as irrelevant and ranked $(C, D) \prec (E, F) \prec (G, H)$ in the ascending order of their relevancy, then the normalized values of the four EDU pairs (A, B), (C, D), (E, F), and (G, H) will be respectively 0, $\frac{1}{3}$, $\frac{2}{3}$, and 1.

To measure median values for ψ_{seg}, ψ_{path} and ψ_{lead} over all candidate EDU pairs of all documents, we group instances (Q, δ_n, δ_s) into n categories of ρ_μ denoted by $\rho_{\mu-n}$. We consider two groups in particular: $n = 2$ (two-valued grading system indicating relevant vs. irrelevant instances), and $n = 4$ (four-valued grading system indicating irrelevant, less relevant, moderately relevant, and highly relevant instances). By considering a coarse two-valued grading and a more finer four-valued grading, we can evaluate the ability of the proposed discourse proximity scores to detect different granularities of relevancies. Table 3 shows the median values of the three discourse proximity scores. We see that for $n = 2$, the EDU pairs ranked as relevant have a smaller median ψ_{seg}, have a smaller median ψ_{path}, and have a smaller median ψ_{lead} in the document. This outcome mirrors the results from [20], where correlations have been found on proximity of query terms in text and document relevance. $n = 4$ case shows similar trends for ψ_{path} and ψ_{lead}. However, for ψ_{seg} we see that EDU pairs ranked as highly relevant have a larger median ψ_{seg} than EDU pairs ranked as less relevant.

We train an RDF classifier as described in Section 4.1, with the test collection as described in Section 5.1, using different combinations of discourse proximity scores as shown in Table 4. In $n = 2$ setting, we train a binary classifier, whereas a multi-class classifier is trained for the $n = 4$ setting. From the leave-one-out F scores shown in Table 4 we see that the combination of ψ_{seg} and ψ_{path} gives the best performance for the $n = 2$ setting, whereas the combination of ψ_{path} and ψ_{lead} gives the best performance for the $n = 4$ setting. In particular, path discourse proximity is found to be a useful feature for detecting relevancy in both settings, which supports our proposal to use discourse trees to represent documents in information retrieval systems.

6 Conclusion

We proposed a discourse search engine that considers the discourse structure in documents to measure the relevance between a query and a document. Three discourse proximity measures that capture different aspects of relevance within the context of a discourse tree were proposed. A random decision forest (RDF) was trained to combine the different discourse proximity scores. We create a test collection for evaluating discourse-based IR systems. Our experiments show the usefulness of the proposed discourse proximity measures. In future, we plan to incorporate the semantic relations between entities in documents within our discourse relevance model; and add TREC collections to the test collection to further improve its performance.

References

1. Baeza-Yates, R.A., Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (1999)
2. Breiman, L.: Random forests. Machine Learning 45(1), 5 – 32 (2001)
3. Brooks, H.M., Belkin, N.J.: Using discourse analysis for the design of information retrieval interaction mechanisms. In: SIGIR. pp. 31–47 (1983)
4. Duc, N.T., Bollegala, D., Ishizuka, M.: Cross-language latent relational search: Mapping knowledge across languages. In: AAAI. pp. 1237 – 1242 (2011)
5. duVerle, D.A., Prendinger, H.: A novel discourse parser based on support vector machine classification. In: ACL. pp. 665–673 (2009)
6. Feng, V.W., Hirst, G.: Text-level discourse parsing with rich linguistic features. In: ACL. pp. 60 – 68 (2012)
7. Hernault, H., Bollegala, D., Ishizuka, M.: A semi-supervised approach to improve classification of infrequent discourse relations using feature vector extension. In: EMNLP. pp. 399 – 409 (2010)
8. Hernault, H., Prendinger, H., duVerle, D., Ishizuka, M.: Hilda: A discourse parser using support vector machine classification. Dialogue and Discourse. An International Journal 1(3), 1–33 (2010)
9. Joty, S., Carenini, G., Ng, R.: A novel discriminative framework for sentence-level discourse analysis. In: EMNLP. pp. 904–915 (2012)
10. Joty, S., Carenini, G., Ng, R., Mehdad, Y.: Combining intra- and multi-sentential rhetorical parsing for document-level discourse analysis. In: ACL. pp. 486–496 (2013)
11. Lafferty, J., McCallum, A., Pereira, F.C.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data (2001)
12. Lan, M., Xu, Y., Niu, Z.: Leveraging synthetic discourse data via multi-task learning for implicit discourse relation recognition. In: ACL. pp. 476–485 (2013)
13. Louis, A., Joshi, A., Nenkova, A.: Discourse indicators for content selection in summarization. In: SIGDIAL. pp. 147 – 156 (2010)
14. Mann, W.C., Thompson, S.A.: Rhetorical structure theory: Toward a functional theory of text organization. Text 8(3), 243 – 281 (1988)
15. Miyao, Y., Ohta, T., Masuda, K., Tsuruoka, Y., Yoshida, K., Ninomiya, T., Tsujii, J.: Semantic retrieval for the accurate identification of relational concepts in massive textbases. In: ACL. pp. 1017–1024 (2006)
16. Sadikov, E., Madhavan, J., Wang, L., Halevy, A.: Clustering query refinements by user intent. In: WWW. pp. 841–850 (2010)
17. Snow, R., O'Connor, B., Jurafsky, D., Ng, A.Y.: Cheap and fast - but is it good? evaluating non-expert annotations for natural language tasks. In: EMNLP. pp. 254 – 263 (2008)
18. Soricut, R., Marcu, D.: Sentence level discourse parsing using syntactic and lexical information. In: NAACL. pp. 149–156 (2003)
19. Subba, R., Eugenio, B.D.: An effective discourse parser that uses rich linguistic information. In: HLT-NAACL. pp. 566–574 (2009)
20. Tao, T., Zhai, C.: An exploration of proximity measures in information retrieval. In: SIGIR. pp. 295–302 (2007)
21. Wang, D.Y., Luk, R.W.P., Wong, K.F., Kwok, K.L.: An information retrieval approach based on discourse type. In: Natural Language Processing and Information Systems, vol. 3999, pp. 197 – 202. Springer Berlin Heidelberg (2006)
22. Zhou, L., Li, B., Gao, W., Wei, Z., Wong, K.F.: Unsupervised discovery of discourse relations for eliminating intra-sentence polarity ambiguities. In: EMNLP. pp. 162–171 (2011)