Comparative Document Analysis for Large Text Corpora

Xiang Ren[†*] Yuanhua Lv[‡] Kuansan Wang[‡] Jiawei Han[†]
[†] University of Illinois at Urbana-Champaign, Urbana, IL, USA
[‡] Microsoft Research, Redmond, WA, USA
[†]{xren7, hanj}@illinois.edu [‡]{yuanhual, Kuansan.Wang}@microsoft.com

ABSTRACT

This paper presents a novel research problem, *Comparative Document Analysis* (*CDA*), that is, *joint* discovery of commonalities and differences between two individual documents (or two sets of documents) in a large text corpus. Given any pair of documents from a (background) document collection, CDA aims to automatically identify sets of quality *phrases* to summarize the commonalities of *both* documents and highlight the distinctions of each *with respect to the other* informatively and concisely. Our solution uses a general graph-based framework to derive novel measures on phrase *semantic commonality* and *pairwise distinction*, where the background corpus is used for computing phrase-document semantic relevance. We use the measures to guide the selection of sets of phrases by solving two joint optimization problems. A scalable iterative algorithm is developed to integrate the maximization of phrase commonality or distinction measure with the learning of phrase-document semantic relevance. Experiments on large text corpora from two different domains—scientific papers and news—demonstrate the effectiveness and robustness of the proposed framework on comparing documents. Analysis on a 10GB+ text corpus demonstrates the scalability of our method, whose computation time grows linearly as the corpus size increases. Our case study on comparing news articles published at different dates shows the power of the proposed method on comparing sets of documents.

1. INTRODUCTION

Comparative text mining is an important problem in text analysis. Identifying common and different content units of various granularity (*e.g.*, words [25], sentences [39], topics [42]) between text items (*e.g.*, document sets [13]) enables effective comparisons among items in a massive corpus. This paper studies a *novel* comparative text mining task, called Comparative Document Analysis (CDA), which leverages *multi-word phrases* (*i.e.*, minimal semantic units) to summarize the commonalities and differences between *two individual documents* (or two sets of documents) by referring to a large background corpus. Given a pair of documents from a large document collection, CDA aims to (1) extract from each document salient phrases and phrase pairs which cover its major content; (2) discover the commonalities between the document pair by selecting salient phrases which are semantically relevant to *both* of them; and (3) find the distinctions for *each* document by selecting salient phrases that are *exclusively* relevant to the document.

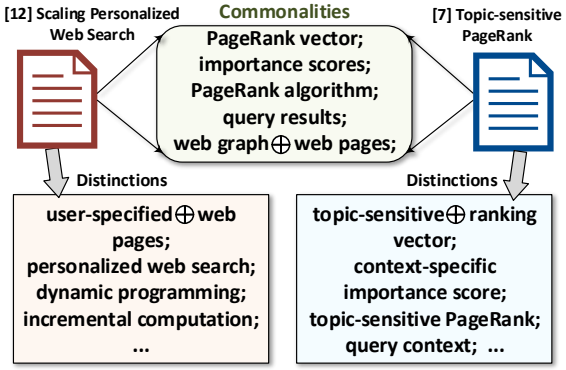


Figure 1: Example output of comparative document analysis for papers [14] and [9]. CDA combines two phrases which frequently co-occur in the documents into a *phrase pair* using the symbol "⊕" (as will be introduced in Sec. 3.1).

With the rapid emergence of massive text-based data in many domains, automatic techniques for CDA have a wide range of applications including social media analysis [36, 23], business intelligence (*e.g.*, customer review analysis [18, 15], news summarization [13, 34, 21]), and scientific literature study (*e.g.*, patentability search [43]). For example, as shown in Fig. 1, a citation recommendation system can show users the common and distinct concepts produced by CDA to help them understand the connections and differences between a query paper [14] and a recommended paper [9]. In a similar way, CDA can reduce the efforts on patentbility searching [43]. To give another example, in product recommendation scenario, CDA can address a user's doubts like "why Amazon recommends *Canon 6D camera* to me after I viewed *Canon 5D3 camera*" by showing common aspects of the two cameras (*e.g.*, "*DSLR camera*", "*full-frame sensor*") and the distinct aspects about *Canon 6D* (*e.g.*, "*WiFi function*"). By analyzing the common/distinct aspects and user's responses, a relevance feedback system can have a better understanding of the user's purchase intent and helps recommend other cameras with similar functions.

Despite of its critical importance in document analysis, CDA is currently done mostly by human efforts and is thus not scalable to large corpora. Our proposed CDA is fully automated, conducts holistic analysis over a large corpus, and does not require domain knowledge (*e.g.*, knowledge bases, domain-specific dictionaries)—it thus can be applied to various domains flexibly. To accomplish this, scalable methods are developed to compute phrase-document

[*]Work done when author was an intern at MSR.

WSDM '17, February 6–10, 2017, Cambridge, United Kingdom.

DOI: http://dx.doi.org/10.1145/3018661.3018690

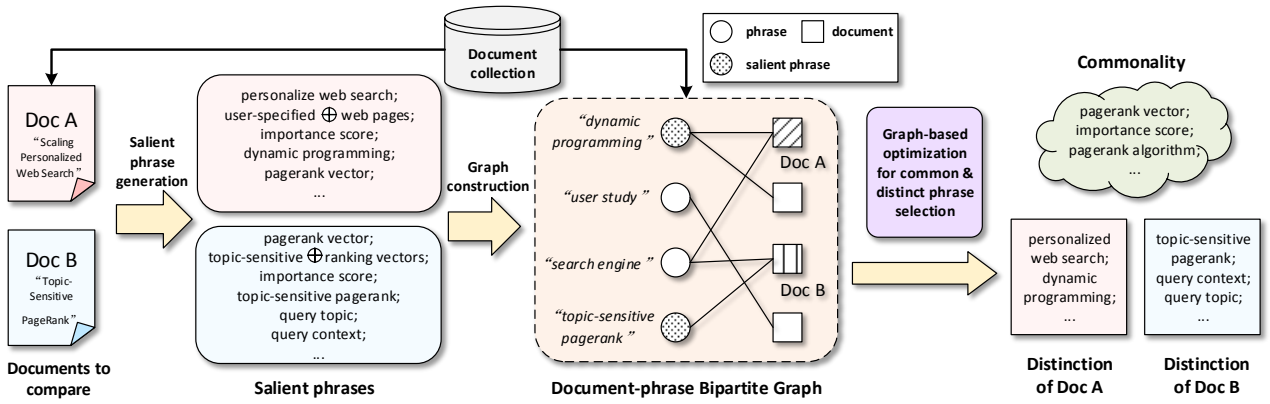


Figure 2: Overview framework of **PhraseCom**.

semantic relevance based on the phrase-document co-occurrences in a large corpus, for measuring phrase commonality and distinction. Moreover, while CDA can be applied to compare either two sets of documents or two individual documents, our study focuses on the latter case: The latter is more challenging since there exist only very limited common content units (*e.g.*, phrase overlap) between individual documents, thus demanding holistic analysis over the large corpus. In addition, many applications do require to compare documents at their finest granularity, at the individual document level instead of document group level, including patentability search and citation recommendation as introduced above.

While there has been some research in comparative text mining, most of these focus on generating *word-based* or *sentence-based distinction* summaries for *sets* of documents. Word-based summarization [25, 42] suffers from limited readability as single words are usually non-informative and bag-of-words representation does not capture the semantics of the original document well—it may not be easy for users to interpret the combined meaning of the words. Sentence-based summarization [13, 39, 20], on the other hand, may be too verbose to highlight the *general* commonalities and differences—users may be distracted by the irrelevant information contained there (as later shown in our case study). Moreover, existing comparative summarization techniques encounter several unique challenges when solving the CDA task.

• **Semantic Commonality:** Commonalities between two documents can intuitively be bridged through phrases that do not *explicitly* occur in both documents but are semantically relevant to both documents. Previous methods consider only content units (*e.g.*, words) explicitly occurring in *both* documents as indication of commonalities but ignore such semantic commonality. The results so generated may suffer from low recall (see our study in Fig. 3).

• **Pairwise Distinction:** Distinct phrases should be extracted based on the pair of compared documents *jointly*—it should be *exclusively* relevant to its own document (*i.e.*, irrelevant to the other document in the pair). Current methods select discriminative content units for each document set *independently*. The phrases so generated thus may not distinguish the two documents effectively.

• **Data Sparsity:** Most existing work relies on word overlap or sentence repetition between the text items in comparison (*i.e.*, document sets) to discover their commonalities and each item's distinctions. However, such sources of evidences may be absent when comparing only two individual documents[1].

We address these challenges with several intuitive ideas. First, to discover semantic commonalities between two documents, we consider phrases which are semantically relevant to *both* documents as *semantic common phrases*, even they do not occur in both documents. Second, to select *pairwise distinct phrases*, we use a novel measure that favors phrases relevant to one document but irrelevant to the other. Third, to resolve data sparsity, we go beyond the simple inter-document content overlap and exploit phrase-document co-occurrence statistics derived from the large background corpus to model semantic relevance between phrases and documents.

To systematically integrate these ideas, we propose a novel graph-based framework called **PhraseCom** (Fig. 2) to unify the formalization of optimization problems on selecting common phrases and distinct phrases. It first segments the corpus to extract candidate phrases where salient phrases for each document are selected based on phrase interestingness and diversity (Sec. 3.1). We then model the semantic relevance between phrases and documents using graph-based propagation over the co-occurrence graphs (Fig. 6) to measure phrase commonality and distinction (Sec. 3.2). We formulate two joint optimization problems using the proposed measures to select sets of common phrases and distinct phrases for a document pair, and present an iterative algorithm to efficiently solve them (Sec. 3.3-3.4). The algorithm tries to integrate the optimizing of the proposed commonality or distinction measure with learning of the semantic relevance scores, and can be flexibly extended to compare two topically-coherent sets of documents (Sec. 3.4). The major contributions of this paper are summarized as follows.

1. We define and study a novel comparative text mining task, comparative document analysis, which uses sets of phrases to jointly represent the commonality and distinctions between a pair (two sets) of documents.
2. We propose a general graph-based framework, PhraseCom, to model the semantic commonality and pairwise distinction for phrases in the documents.
3. We formulate joint optimization problems to integrate the maximization of phrase commonality or distinction with the learning of phrase-document semantic relevance, and develop an efficient algorithm for solving them.
4. Experiments on datasets from different domains—news and academic papers—demonstrate that the proposed method achieves significant improvement over the state-of-the-art (*e.g.*, a 56% enhancement in F1 score on the Academia dataset over the next best compared method).

2. PROBLEM DEFINITION

The input to comparative document analysis is a collection of documents $\mathcal{D} = \{d_1, d_2, \ldots, d_n\}$, a set of document pairs $\mathcal{U} = \{(d, d')\}_{d,d' \in \mathcal{D}}$ from $\mathcal{D}$ for comparison, and a set of positive example phrases $\mathcal{P}^+$ collected from Wikipedia article titles for candidate phrase mining (as later introduced in Sec. 3.1).

[1]One may argue that we can expand each individual document into a *pseudo document* by retrieving its topic-related documents. However, this is expensive in both computation (as it takes each document as a query to hit the collection) and storage.

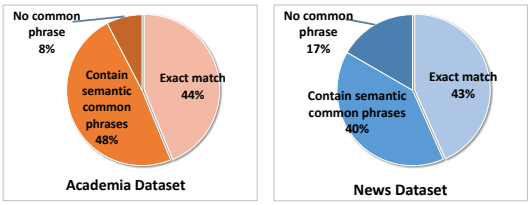


Figure 3: Study on semantic commonality: we show %document pairs that contain *semantic common phrases* (*i.e.*, phrase which does not occur in *both* document but represents the commonality) on the two evaluation sets. We found 62.37% of the gold standard common phrases in evaluation sets are semantic common phrases.

Notations. A *phrase*, p, is a single-word or multi-word sequence in the text document which represents a cohesive content unit (*e.g.*, a noun phrase like "*automatic text summarization*", or a verb phrase like "*earthquake struck*"). We further consider combining two phrases p_a and p_b into a *phrase pair*, *i.e.*, $p_a \oplus p_b$, if they tend to co-occur with each other frequently in the document[2] (*e.g.*, "*web graph*⊕*web pages*" in Fig. 1). Let $\mathcal{P} = \{p_1, \ldots, p_m\}$ denote the m unique phrases extracted from the corpus $\mathcal{D}$. For a pair of documents $(d, d') \in \mathcal{U}$, we denote the two sets of salient phrases extracted from them as $\mathcal{S}$ and $\mathcal{S}'$, respectively. A binary vector $\mathbf{y}^c \in \{0,1\}^m$ is used to indicate whether salient phrases from the set $\mathcal{S} \cup \mathcal{S}'$ are selected to form the set of common phrases $\mathcal{C}$. Another two binary vectors, $\mathbf{y}, \mathbf{y}' \in \{0,1\}^m$, are used to indicate whether phrases from $\mathcal{S}$ and $\mathcal{S}'$ are selected to form the set of distinct phrases for d and d', respectively (denoted as $\mathcal{Q} \subseteq \mathcal{S}$ and $\mathcal{Q}' \subseteq \mathcal{S}'$).

Problem. By estimating $\{\mathbf{y}^c, \mathbf{y}, \mathbf{y}'\}$ with the constraint that $\mathcal{C} \cap \mathcal{Q} = \mathcal{C} \cap \mathcal{Q}' = \varnothing$, one can generate the three sets of phrases, *i.e.*, $\mathcal{C} = \{p \mid p \in \mathcal{S} \cup \mathcal{S}',\ y^c_p = 1\}$, $\mathcal{Q} = \{p \mid p \in \mathcal{S},\ y_p = 1\}$, and $\mathcal{Q}' = \{p \mid p \in \mathcal{S}',\ y'_p = 1\}$. Formally, we define the problem of comparative document analysis (CDA) as follows.

DEFINITION 1 (PROBLEM DEFINITION). ***Given*** *a document collection $\mathcal{D}$, a set of document pairs $\mathcal{U}$ and a set of positive example phrases $\mathcal{P}^+$, CDA* ***aims to****: (1) extract salient phrases $\mathcal{S}$ for each document $d \in \mathcal{D}$; and (2) for each pair of comparing documents $(d, d') \in \mathcal{U}$, estimate the indicator vectors $\{\mathbf{y}^c, \mathbf{y}, \mathbf{y}'\}$ for phrases to predict the comparison result sets $\{\mathcal{C}, \mathcal{Q}, \mathcal{Q}'\}$.*

Non-goals. In our study, we assume the given document pairs in $\mathcal{U}$ are comparable, *i.e.*, they share some common aspects or belong to the same topic. For example, they can be two news articles (or scientific papers) on similar topics. Such document pairs can come from document retrieval and item recommendation results. It is not the focus of this paper to generate such document pairs.

3. COMPARATIVE DOCUMENT ANALYSIS

Overall framework of PhraseCom (see Fig. 2) is as follows:

1. Perform distantly-supervised phrase segmentation on $\mathcal{D}$ and select salient phrases $\mathcal{S}$ for $d \in \mathcal{D}$ by optimizing both phrase interestingness and diversity. (Sec. 3.1).
2. Construct a phrase-document co-occurrence graph to help model semantic relevance as well as to assist measuring phrase commonality and distinction (Sec. 3.2).
3. Estimate indicator vectors $\{\mathbf{y}^c, \mathbf{y}, \mathbf{y}'\}$ for the three sets $\{\mathcal{C}, \mathcal{Q}, \mathcal{Q}'\}$ by solving the two proposed optimization problems with the efficient algorithms. (Sec. 3.3-3.4).

Each step will be elaborated in the following sections.

[2]Without explicitly mentioning, we refer both phrase and phrase pair as phrase in the rest of the paper.

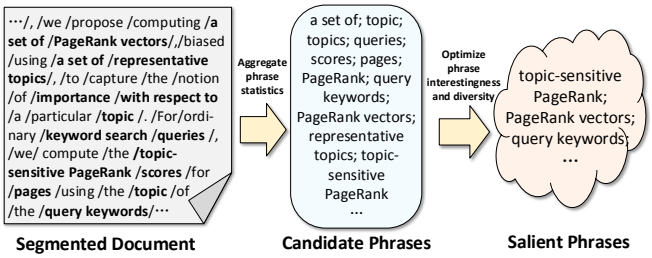


Figure 4: Output for document [9] in salient phrase generation.

3.1 Salient Phrase Generation

To ensure the generation of cohesive, informative, salient phrases for each document, we introduce a scalable, data-driven approach by incorporating both local syntactic signals and corpus-level statistics (see Fig. 4 with examples). Our method first uses a phrase mining algorithm to partition the text into non-overlapping segments (*i.e.*, candidate phrases). Then it adopts both *interestingness* and *diversity* measures in a joint optimization framework to guide the filtering of low importance phrases and removing redundant phrases.

Candidate Phrase Mining. Given a word sequence, output of candidate phrase mining is a sequence of phrases each representing a cohesive content unit (see Fig. 4). Our work relies on existing phrase mining methods (data-driven [22, 1] or linguistic [44]) to extract candidate phrases but we do not address their limits here. In this study, we adopt a data-driven phrase segmentation algorithm, SegPhrase [22], which uses distant supervision in conjunction with Wikipedia for training[3] and thus does not rely on human-annotated data. To enhance readability, we combine the candidate phrases which have good concordance (*i.e.*, p_a and p_b co-occur more than 3 times in a window of 10 words in d) into phrase pairs $p_a \oplus p_b$.

Salient Phrase Selection. After mining candidate phrases, each document can be seen as a *bag of phrases*. However, the majority of candidate phrases are not representative for the document. To select salient phrases for a document, we consider two different aspects to measure the phrase salience, *i.e.*, phrase *interestingness* and phrase *diversity*. The intuition behind phrase interestingness is simple [1]: a phrase is more interesting to the document if it appears frequently in the current document while relatively infrequently in the entire corpus. Let $\mathcal{P}_d$ denote the set of phrases from the segmented document d, $n(p,d)$ denote the frequency of p in d, and $n(p,\mathcal{D})$ denote the document frequency of p in $\mathcal{D}$. The interestingness measure $r(\cdot)$ of p in $d \in \mathcal{D}$ is defined as follows [1].

$$r_{\mathcal{D}}(p,d) = \Big(0.5 + \frac{0.5 \times n(p,d)}{\max_{t \in \mathcal{P}_d} n(t,d)}\Big)^2 \cdot \log\Big(\frac{|\mathcal{D}|}{n(p,\mathcal{D})}\Big), \quad (1)$$

which is the product of the square of normalized term frequency and the inverse document frequency. Interestingness score of phrase pair $p_a \oplus p_b$ is computed using an intra-document point-wise mutual information and is discount by its document frequency as follows.

$$r_{\mathcal{D}}(p_a \oplus p_b, d) = \frac{\frac{n(p_a \oplus p_b, d)}{|\mathcal{P}_d|}}{\frac{n(p_a,d)}{|\mathcal{P}_d|}\frac{n(p_b,d)}{|\mathcal{P}_d|}} \cdot \log\Big(\frac{|\mathcal{D}|}{n(p_a \oplus p_b, \mathcal{D})}\Big). \quad (2)$$

Interestingness scores for both phrases and phrase pairs from a document are normalized into $[0,1]$ for comparison.

To impose good diversity on the set of selected phrases, we require them to be different from each other. We adopt *Levenshtein similarity* to measure the string similarity $M(p_a, p_b)$ between two phrases p_a and p_b. One can also apply semantic similarity measures

[3]Following the procedure introduced in [22], we randomly select 500 Wikipedia article titles to form the set of positive example phrases $\mathcal{P}^+$.

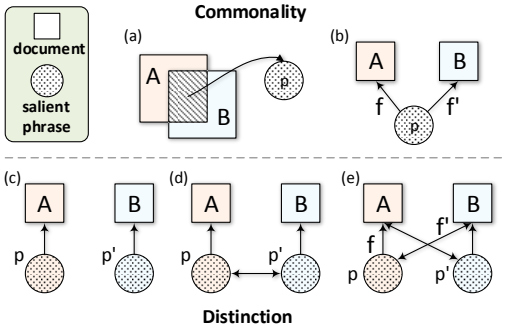


Figure 5: An illustration of the proposed measures. (a) Intersection model; (b) Semantic commonality model (ours): phrase is relevant to *both* documents; (c) Independent distinction model; (d) Joint distinction model: similar phrases are removed; (e) Pairwise distinction model (ours): phrase is *exclusively relevant* to one document.

like distributional similarity [8]. To select a subset $\mathcal{S} \subset \mathcal{P}_d$ of K salient phrases for document d, we solve an optimization problem to maximize interestingness and diversity *jointly* as follows.

$$\underset{\mathcal{S}\subset\mathcal{P}_d,\ |\mathcal{S}|=K}{\operatorname{argmax}} \mathcal{H}(\mathcal{S}) = \mu \sum_{p_a\in\mathcal{S}} q_a r_a - \sum_{p_a,p_b\in\mathcal{S}} r_a M_{ab} r_b, \quad (3)$$

where $r_i = r_{\mathcal{D}}(p_i, d)$ is the interestingness score, $M_{ij} = M(p_i, p_j)$ is phrase similarity score, and $q_a = \sum_{j=1}^{|\mathcal{P}_d|} M_{aj} r_j$ is the weight for p_a. The first term is overall interestingness of $\mathcal{S}$. If two phrases are equally interesting, it flavors the one that comes from a big cluster (*i.e.*, the phrase is similar to many other phrases in $\mathcal{P}_d$). The second term measures the similarity among the phrases within $\mathcal{S}$. That is, it penalizes the selection of multiple phrases which are very similar to each other. A near-optimal solution of Eq. (3) can be obtained by an efficient algorithm [10] with time cost $\mathcal{O}(|\mathcal{P}_d|^2 + |\mathcal{P}_d|K)$.

3.2 Commonality and Distinction Measures

To generate $\{\mathcal{C}, \mathcal{Q}, \mathcal{Q}'\}$ from the salient phrase sets $\mathcal{S}$ and $\mathcal{S}'$, one solution [25] is to use salient phrase occurring in both documents to represent commonalities and the remaining salient phrases to highlight the distinctions (*i.e.*, (a) and (c) in Fig. 5). However, such a solution ignores semantic common phrases, cannot guarantee the pairwise distinction property, and may include overly specific yet less informative phrases (*e.g.*, "*partial vectors*" in Fig. 6). An alternative solution is to cluster salient phrases from both documents to identify "commonality clusters" and "distinction clusters". However, it is non-trivial to decide cluster granularity (*i.e.*, number of clusters) as it varies for different document pairs, and is hard to capture pairwise distinction property. To resolve these issues, we derive semantic relevance between phrases and documents based on their corpus-level co-occurrences statistics (Sec. 3.3), and formalize novel objectives to measure semantic commonality and pairwise distinction for phrases. Ideally, a good common phrase should be *semantically* relevant to *both* documents; a good distinct phrase should be *relevant* to this document but *irrelevant* to the other one; and a good phrase should have reasonable popularity in the corpus.

Commonality Measure. Let function $f(p, d) : \mathcal{P} \times \mathcal{D} \mapsto \mathbb{R}_0^+$ denote the relevance score between $p \in \mathcal{P}$ and $d \in \mathcal{D}$, (will be elaborated later in Sec. 3.3). We define the commonality score function $\Phi(p, d, d') : \mathcal{P} \times \mathcal{D} \times \mathcal{D} \mapsto \mathbb{R}_0^+$ to measure how well phrase p can represent the commonality between the document pair (d, d'). The following hypothesis guides our modeling of commonality score.

HYPOTHESIS 1 (PHRASE COMMONALITY). *Given document pair (d, d') for comparison, phrase p tends to have high commonality score $\Phi(p, d, d')$ if and only if the relevance scores between the phrase and both documents, i.e., $f(p, d)$ and $f(p, d')$, are high.*

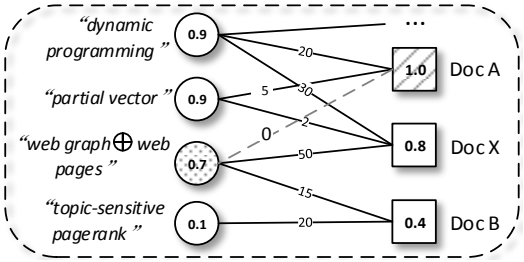


Figure 6: Illustration of relevance scores to Doc A derived from the constructed bipartite graph. Our method can infer the relevance between "*web graph⊕web pages*" and Doc A (*i.e.*, 0.7) even it does not occur in Doc A.

In Fig. 6, for example, "*web graph⊕web pages*" has high commonality score since it has high relevance score to both Doc B (it occurs frequently in Doc B) and Doc A (it occurs frequently in Doc X and Doc X contains several phrase that are relevant to Doc A). Formally, we define the *commonality score function* as follows.

$$\Phi(p, d, d') = \ln\Big(1 + f(p, d) \cdot f(p, d')\Big). \quad (4)$$

Similar to the product-of-experts model [11], it models the commonality score as the product of the two relevance scores, likes an "and" operation. An alternative definition for $\Phi(\cdot)$ is the summation of two relevance scores, *i.e.*, $\Phi(p, d, d') = f(p, d) + f(p, d')$. However, as an "or" operation, the score so modeled may be dominated by the larger relevance score among the two. For instance, the case $f(p, d) = f(p, d') = 0.5$ and the case $f(p, d) = 0.1, f(p, d') = 0.9$ share the same commonality score, but the former represents better commonality. We compare these two models in our experiments.

Distinction Measure. A good phrase for highlighting the distinction of d between (d, d') should not only distinguish d from d' but also have good readability, *i.e.*, not overly specific. For example, "*dynamic programming*" in Fig. 6 serves as a good distinct phrase for Doc A, when comparing with Doc B—it has strong association with Doc A and weak association with Doc B, and is fairly popular in the corpus. On the other hand, "*partial vector*" has similar association pattern with Doc A and Doc B but it is rarely mentioned in the corpus, *i.e.*, overly specific. We use a distinction score function $\Pi(p, d, |d') : \mathcal{P} \times \mathcal{D} \times \mathcal{D} \mapsto \mathbb{R}$ to measure how well p can highlight the distinction of d from d' based on the following hypothesis.

HYPOTHESIS 2 (PHRASE DISTINCTION). *Given (d, d'), phrase p tends to have high distinction score $\Pi(p, d|d')$ if it has relatively higher relevance score $f(p, d)$ to document d compared with its relevance score $f(p, d')$ to document d'.*

Specifically, we define $\Pi(p, d|d')$ based on the division of relevance to d by the relevance to d', which has the following form.

$$\Pi(p, d|d') = \ln\Big(\frac{f(p, d) + \gamma}{f(p, d') + \gamma}\Big). \quad (5)$$

Here, a smoothing parameter $\gamma = 1$ is used to avoid selecting phrase p with too small $f(p, d)$ or $f(p, d')$. In particular, the relevance score $f(p, d)$ incorporates the popularity of phrase p in the collection (see Sec. 3.3) and thus can filter overly specific phrases. A phrase will receive high distinct score in two cases: (1) p has high relevance score to d and moderate or low relevance score to d'; and (2) p has moderate relevance score to d and low relevance score to d'. The second case helps include more phrases to represent the distinctions even they are moderately relevant to its own document. An alternative way to define the distinction score is by score difference, *i.e.*, $\Pi(p, d|d') = f(p, d) - f(p, d')$. Such score functions prefer the first case than the second one, and thus will suffer from low recall. We compare with this alternative measure in the experiments.

$d,\ \mathcal{D} = \{d_j\}_{j=1}^{n}$	Document, text corpus (size n)
$\mathcal{U} = \{(d, d')\}$	Document pairs for comparison
$p,\ p_a \oplus p_b$	Phrase, phrase pair
$\mathcal{P} = \{p_i\}_{i=1}^{m}$	Unique phrases (pairs) in $\mathcal{D}$ (size m)
$\mathcal{S},\ \mathcal{S}'$	Salient phrases extracted from d, d'
$\mathcal{C}$	Common phrase set of document pair (d, d')
$\mathbf{y^c} \in \{0,1\}^m$	Binary indicator vector for $\mathcal{C}$
$\mathcal{Q},\ \mathcal{Q}'$	Distinct phrase sets of d, d'
$\mathbf{y}\ , \mathbf{y}' \in \{0,1\}^m$	Binary indicator vectors for $\mathcal{Q}$, $\mathcal{Q}'$

Table 1: Notations.

3.3 Comparative Selection Optimization

With the proposed measures, we now are concerned of the following two questions: (1) how to derive phrase-document relevance score $f(p, d)$; and (2) how to select the phrase sets $\{\mathcal{C}, \mathcal{Q}, \mathcal{Q}'\}$ for a document pair. To answer these two questions, we formulate optimization problems to jointly learn phrase-document semantic relevance on a constructed graph, and select common/distinct phrases by maximizing the corresponding measures.

Graph-Based Semantic Relevance. A simple idea to compute $f(p, d)$ is to use bag-of-words similarity measures such as BM25 score. However, the score so generated may not capture the semantic relatedness between them (see CDA-NoGraph in Sec. 4). Our solution leverages graph-based semi-supervised learning [45, 46] to model the semantic relevance between phrases and documents, and further integrates the relevance learning with phrase selection in a mutually enhancing way. It treats the target document $d \in \mathcal{D}$ as positive label and tries to rank the phrases $\mathcal{P}$ and the remaining documents $\mathcal{D} \setminus d$ based on the intrinsic structure among them.

By exploiting the aggregated co-occurrences between phrases and their supporting documents (*i.e.*, documents where the phrase occurs) across the corpus, we weight the importance of different phrases for a document, and use their connected edge as bridges to propagate the relevance scores between phrases and documents.

HYPOTHESIS 3 (DOCUMENT-PHRASE RELEVANCE). *If a phrase's support documents are relevant to the target document, then the phrase tends to be relevant to the target document; If a document contains many phrases that are relevant to the target document, the document is likely relevant to the target document.*

In Fig. 6, for example, if we know "*dynamic programming*" and "*partial vector*" have high relevance scores regarding the target document Doc A, and find that the two phrases have strong association with Doc X, then Doc X is likely relevant to Doc A. This may reinforce the relevance score propagation that "*web graph⊕web page*" is also relevant to Doc A, if the other support documents (*e.g.*, Doc B) are also relevant to Doc A.

Specifically, we construct a bipartite graph G to capture the co-occurrences between all the phrases $\mathcal{P}$ and documents $\mathcal{D}$. A bi-adjacency matrix $\mathbf{W} \in \mathbb{R}_0^{+m \times n}$ is used to represent the edge weights for the links where W_{ij} is the BM25 score [26] ($k_1 = 1.2$, $b = 0.75$) between $p_i \in \mathcal{P}$ and $d_j \in \mathcal{D}$ if p_i occurs in d_j; and zero otherwise. We use function $g(d_a, d_b) : \mathcal{D} \times \mathcal{D} \mapsto \mathbb{R}_0^+$ to denote the relevance score between any two documents (d_a, d_b), and define vectors $\mathbf{f} \in \mathbb{R}_0^{+m}$ as $f_i = f(p_i, d)$ and $\mathbf{g} \in \mathbb{R}_0^{+n}$ as $g_j = g(d_j, d)$. Following Hypothesis 3, we use target document d as positive label, and model the phrase-document relevance score propagation by combining a graph regularization term and a supervision term [45].

$$\mathcal{L}_{d,\alpha}(\mathbf{f}, \mathbf{g}) = \sum_{i=1}^{m}\sum_{j=1}^{n} W_{ij}\Big(\frac{f_i}{\sqrt{D_{ii}^{(\mathcal{P})}}} - \frac{g_j}{\sqrt{D_{jj}^{(\mathcal{D})}}}\Big)^2 + \alpha\|\mathbf{g} - \mathbf{g}^0\|_2^2.$$

Here, we define the indicator vector $\mathbf{g}^0 \in \{0,1\}^n$ to impose the positive label of d in the second term, where $g_d^0 = 1$ and 0 otherwise. A tuning parameter $\alpha > 0$ is used to control the strength of supervision from d on the score propagation. Moreover, we normalize $\mathbf{W}$ by the popularity of the phrases and documents to reduce the impact of overly popular phrases [46], using node degrees $D_{ii}^{(\mathcal{P})} = \sum_{j=1}^{n} W_{ij}$ and $D_{jj}^{(\mathcal{D})} = \sum_{i=1}^{m} W_{ij}$ in the first term. The proposed framework can also incorporate external knowledge on phrases into the constructed graph, such as semantic relations between phrases in knowledge bases [2] and phrase similarity computed using word embeddings [28]. We leave this as future work.

The Joint Optimization Problems. Relevance score $f(p, d)$ can be directly learned by minimizing $\mathcal{L}_d(\mathbf{f}, \mathbf{g})$ but the score so computed is *document independent*—it considers information from d while ignoring that from d' when comparing (d, d'). To address this, we propose two joint optimization problems for selecting common phrases and distinct phrases, respectively, by incorporating information from both documents. The intuition is simple: phrases which are likely common (distinct) phrases can reinforce the propagation between relevance scores by serving as extra positive labels (*i.e.*, as complement to d). In our experiments, we compare with the independent optimization method (*i.e.*, CDA-TwoStep).

Formally, to discover the common phrases $\mathcal{C}$ between a document pair $(d, d') \in \mathcal{U}$, we propose the common phrase selection problem which unifies two different objectives: (i) selection of $\mathcal{C} \subset \mathcal{S} \cup \mathcal{S}'$ to maximize the overall commonality score; and (ii) minimization of the graph-based regularization terms to learn relevance scores. Let $\mathbf{f}' \in \mathbb{R}_0^{+m}$ denote phrase-document relevance scores for $\mathcal{P}$ with $f_i' = f(p_i, d')$, and $\mathbf{g}' \in \mathbb{R}_0^{+n}$ denote document-document relevance scores for $\mathcal{D}$ with $g_j' = g(d_j, d')$. The common phrase selection problem is formulated as follows.

$$\begin{aligned}\min_{\mathbf{y}^c, \mathbf{f}, \mathbf{f}', \mathbf{g}, \mathbf{g}'} \mathcal{O}_{\alpha,\lambda} = &-\lambda \sum_{i=1}^{m} y_i^c \cdot \Phi(p_i, d, d') \\ &+ \frac{1}{2}\mathcal{L}_{d,\alpha}(\mathbf{f}, \mathbf{g}) + \frac{1}{2}\mathcal{L}_{d',\alpha}(\mathbf{f}', \mathbf{g}') \\ \text{s.t.}\quad y_i^c \cdot \Phi(p_i, d, d') &\geq y_i^c \sum_{p_j \in \mathcal{S}} \Phi(p_j, d, d')/|\mathcal{S}|,\ \ \forall p_i \in \mathcal{P}; \\ y_i^c \cdot \Phi(p_i, d, d') &\geq y_i^c \sum_{p_j \in \mathcal{S}'} \Phi(p_j, d, d')/|\mathcal{S}'|,\ \ \forall p_i \in \mathcal{P};\end{aligned} \tag{6}$$

The first term in objective $\mathcal{O}$ aggregates the commonality scores for the selected phrases, and the tuning parameter $\lambda > 0$ controls the trade-off between it and the second and third terms which model the relevance score propagation on graph. We add the first and second constraints to automatically decide the size of $\mathcal{C}$, by enforcing that the selected phrases should have higher commonality score than the average commonality scores computed over salient phrases $\mathcal{S}$ and $\mathcal{S}'$, respectively. We also enforce $\mathcal{C} \subset \mathcal{S} \cup \mathcal{S}'$ when solving Eq. (6).

To jointly select the distinct phrases $\{\mathcal{Q}, \mathcal{Q}'\}$ for pair $(d, d') \in \mathcal{U}$, we propose the distinct phrase selection problem. It aims to: (i) select phrases $\mathcal{Q} \subset \mathcal{S}$ and $\mathcal{Q}' \subset \mathcal{S}'$ to maximize the overall distinction scores; and (ii) derive relevance scores by minimizing the graph-based regularization terms jointly.

$$\begin{aligned}\min_{\mathbf{y}, \mathbf{y}', \mathbf{f}, \mathbf{f}', \mathbf{g}, \mathbf{g}'} \mathcal{F}_{\alpha,\lambda} = &-\lambda \sum_{i=1}^{m} \Big[y_i \cdot \Pi(p_i, d|d') + y_i' \cdot \Pi(p_i, d'|d)\Big] \\ &+ \frac{1}{2}\mathcal{L}_{d,\alpha}(\mathbf{f}, \mathbf{g}) + \frac{1}{2}\mathcal{L}_{d',\alpha}(\mathbf{f}', \mathbf{g}') \\ \text{s.t.}\quad y_i \cdot \Pi(p_i, d|d') &\geq y_i \sum_{p_j \in \mathcal{S}} \Pi(p_j, d|d')/|\mathcal{S}|,\ \ p_i \in \mathcal{P}; \\ y_i' \cdot \Pi(p_i, d'|d) &\geq y_i' \sum_{p_j \in \mathcal{S}'} \Phi(p_j, d'|d)/|\mathcal{S}'|,\ \ p_i \in \mathcal{P};\end{aligned} \tag{7}$$

The first term represents the aggregated distinction score over the two sets of selected distinct phrases, *i.e.*, $\mathcal{Q}$ and $\mathcal{Q}'$, respectively. Similarly, the above two constraints help control the size of the

result sets. Moreover, we impose the constraints that $\mathcal{Q} \subset \mathcal{S},\ \mathcal{Q}' \subset \mathcal{S}'$ and $\mathcal{C} \cap \mathcal{Q} = \mathcal{C} \cap \mathcal{Q}' = \emptyset$ (given set $\mathcal{C}$) when solving Eq. (7).

3.4 An Efficient Algorithm

The optimization problems in Eqs. (6) and (7) are mix-integer programming and thus are NP-hard. We propose an approximate solution for each problem based on the alternative minimization framework [37]: first estimate $\{\mathbf{f}, \mathbf{f}', \mathbf{g}, \mathbf{g}'\}$ through minimizing $\mathcal{O}$ (or $\mathcal{F}$) while fixing $\mathbf{y}^c$ (or $\{\mathbf{y}, \mathbf{y}'\}$); then fix $\{\mathbf{f}, \mathbf{f}', \mathbf{g}, \mathbf{g}'\}$ and optimize $\mathcal{O}$ (or $\mathcal{F}$) with respect to $\mathbf{y}^c$ (or $\{\mathbf{y}, \mathbf{y}'\}$) by imposing the constraints; and iterate between these two steps until reaching the convergence of the objective functions $\mathcal{O}$ and $\mathcal{F}$ (*i.e.*, outer loop).

Specifically, to estimate $\{\mathbf{f}, \mathbf{f}', \mathbf{g}, \mathbf{g}'\}$, we take derivative on $\mathcal{O}$ (or $\mathcal{F}$) with respect to each of the variables in $\{\mathbf{f}, \mathbf{f}', \mathbf{g}, \mathbf{g}'\}$ while fixing other variables; obtain the update rules by setting the derivatives to zero; and iteratively update between $\{\mathbf{f}, \mathbf{f}', \mathbf{g}, \mathbf{g}'\}$ until the reconstruction error $\mathcal{L}_{d,\alpha}$ converges (*i.e.*, inner loop). With the updated relevance scores, we then compute the commonality (or distinction) scores for all $p_i \in \mathcal{P}$, and update the indicator vectors $\mathbf{y}^c$ (or $\{\mathbf{y}, \mathbf{y}'\}$) by checking whether the estimated scores satisfy the constraints in Eqs. (6) and (7), *i.e.*, y_i^c (or $y_i,\ y_i'$) is set as 1 if p_i's relevance scores satisfy the constraints; and set as 0 otherwise.

For convergence analysis, the proposed algorithm applies block coordinate descent on problems in Eqs. (6) and (7). The proof procedure in [37] (not included for lack of space) can be adopted to prove convergence for PhraseCom (to the local minima).

Computational Complexity Analysis. Suppose that corpus $\mathcal{D}$ have n documents and $N_{\mathcal{D}}$ words, and that the number of candidate phrases extracted from a document is bounded by a constant. The time cost of data preparation (*i.e.*, salient phrase generation and graph construction) is $\mathcal{O}(N_{\mathcal{D}})$. The time cost of comparative selection optimization on a document pair is $\mathcal{O}(n)$. In practice, data preparation can be done in advance, and reused in comparative selection optimization for different document pairs. Given t document pairs, the total time cost for PhraseCom is $\mathcal{O}(N_{\mathcal{D}} + nt)$, which is linear to $N_{\mathcal{D}}$, n and t. Furthermore, comparative selection optimization on different document pairs could be easily parallelized as the nature of independence between document pairs.

Extension to compare two sets of documents. PhraseCom can be easily extended to compare two sets of documents, *i.e.*, $(\mathcal{D}_a, \mathcal{D}_b)$. Let $\mathcal{S}_a = \cup_{d \in \mathcal{D}_a} \mathcal{S}_d$ and $\mathcal{S}_b = \cup_{d \in \mathcal{D}_b} \mathcal{S}_d$ denote the salient phrases extracted from the two document sets, respectively. Our method can directly replaces $\{\mathcal{S}, \mathcal{S}'\}$ by $\{\mathcal{S}_a, \mathcal{S}_b\}$, initializes the vectors $\{\mathbf{g}^0, \mathbf{g}'^0\}$ based on $\{\mathcal{D}_a, \mathcal{D}_b\}$, and derive the comparison results $\{\mathcal{C}, \mathcal{Q}_a, \mathcal{Q}_b\}$ following the aforementioned optimization procedure.

4. EXPERIMENTS

4.1 Data Preparation

Our experiments use two real-world datasets[4]:

- **Academia:** We collected 205,484 full-text papers (158M tokens and 1.98M unique words) published in a variety of venues between 1990 and 2015 from ACM Digital Library.
- **News:** constructed by crawling news articles published between Mar. 11 and April 11, 2011 with keywords "Japan Tsunami" from NewsBank. This yields a collection of 67,809 articles (44M tokens and 247k unique words).

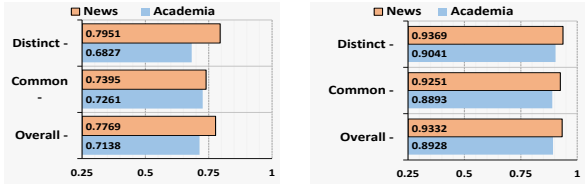
Salient Phrases. For phrase segmentation, we set maximal pattern length to 5, minimum support to 10, and non-segmented ratio to 0.05 in the SegPhrase algorithm (as used in [22]) to extract candidate phrases from the corpus. We then used the GenDeR algorithm in [10] to solve the salient phrase selection problem in Eq. (3). We set weighting parameter $\mu = 3$ and maximal number of salient phrase selected for each document (*i.e.*, K) as 30, after tuning on the validation sets. Member phrases in salient phrase pairs were removed from the salient phrase set to avoid redundancy.

[4] http://dl.acm.org/; http://www.newsbank.com/

Data sets	Academia	News
#Documents	205,484	67,809
#Candidate phrases	611,538	224,468
#Salient phrases	316,194	119,600
#Salient phrase pairs	204,744	11,631
#Unique words	1.98M	246,606
#Links	153.19M	2.85M
#Salient phrases (pairs) per doc	18.95 (2.42)	17.43 (1.31)

Table 2: Statistics of the datasets.

Bipartite Graphs. We followed the introduction in Sec. 3.3 to construct the phrase-document bipartite graph for each dataset. To compute the BM25 score between a phrase pair and a document, we concatenated the two member phrases in the pair together. Table 2 summarizes the statistics of the two constructed graphs.



(a) Kappa coefficient (b) Observed agreement ratio

Figure 7: Inter-annotator agreement.

Evaluation Sets. We selected papers published in two different areas (*i.e.*, KDD and SIGIR) as well as news articles about Japan Tsunami to construct three evaluation sets. To generate document pairs $\mathcal{U}$, we computed the document cosine similarity using TF-IDF vectors. We sampled 350 pairs of *more related* documents (score > 0.6), 350 pairs of *less related* documents ($0.05 <$ score < 0.2) and 350 *random* pairs for each evaluation set. As the number of phrases in each document is large, we adopted the *pooling* method [26] to generate gold standard phrases. For each document pair, we constructed the pool with the results returned by all the compared methods. To further increase the coverage, we also added all the salient words generated by WordMatch [25]. Human assessment was conducted by three computer science researchers. A phrase is annotated as "perfect" (2) distinct (common) phrase if it can connect (distinguish) two documents and is not overly specific or general; as "good" (1) distinct (common) phrase if it is general but still can distinguish (connect) two documents; and as "bad" (0) otherwise. This yields 85k, 86k and 59k annotated phrases and words for KDD, SIGIR and News evaluation sets, respectively. To evaluate human assessment agreement among the annotators, Fig. 7 summarizes the average kappa values (a statistic that measures inter-judge agreement) and relative observed agreement ratios (% of items for which the two annotators' evaluations agree) on the datasets. The results (*e.g.*, 0.7+ kappa values and ∼0.9 agreement ratios) demonstrate that the human annotators have good inter-judge agreement on both common and distinct phrases.

4.2 Experimental Settings

In our testing of PhraseCom and its variants, we set $\{\lambda, \alpha\} = \{0.1, 100\}$ based on the required condition and effectiveness study on a validation set. Empirically, our performance does not change

dramatically across a wide choices of parameters. For convergence criterion, we stop the outer (inner) loops in the algorithm if the relative changes of $\mathcal{O}$ in Eq. (6) and $\mathcal{F}$ in Eq. (7) (reconstruction error $\mathcal{L}_{d,\alpha}$) are smaller than 10^{-4}.

Compared Methods: We compared the proposed method with its variants which only model part of the proposed hypotheses. Several state-of-the-art comparative summarization methods were also implemented (parameters were first tuned on our validation sets): (1) **WordMatch** [25]: extracts top-N *salient words* based on TF-IDF scores ($N = 20$ after tuning on the validation sets). It generates common set based on (a) in Fig. 5 and takes the rest as distinct sets; (2) **TopicLabel** [27]: TopicLabel selects salient phrases based on first-order relevance measure and uses same method as WordMatch to generate common and distinct sets; (3) **PatentCom** [43]: a state-of-the-art graph-based comparative summarization method. We adopt its common and distinct phrase sets for comparisons; (4) **StringFuzzy**: It follows Sec. 3.1 to extract salient phrases. It finds common phrase if its BM25 scores to *both* documents are larger than a threshold (set as 3.0 after tuning on the validation sets) and uses the remaining salient phrases to form distinct sets; (5) **ContextFuzzy**: Different from StringFuzzy, it measures cosine similarity between the pseudo-document of a phrase (formed by all contexts of the phrase in a 10 words window in the corpus) and a document; (6) **Word2Vec-Clus**: It learns embeddings for salient phrases from $\mathcal{D}$ using the skip-gram model[5] [28], and then clusters phrases using X-means algorithm[6] [30] which decides the number of clusters from $[5, 500]$ automatically. For clusters whose phrases occur in both documents, we add the phrases that are closest to each cluster centroid to the common set. For clusters whose phrases occur in only one document, we add the phrases closest to each cluster centroid to the corresponding distinct sets; and (7) **NMF-Clus**: Different from Word2Vec-Clus, it derives embeddings for salient phrases by doing NMF [19] on the graph G. Dimensionality of the embeddings is set as 200 after tuning on the validation set.

For PhraseCom, besides the proposed full-fledged model, **CDA**, we also compare with its variants which implement our intuitions differently: (1) **CDA-NoGraph**: It uses BM25 scores to measure phrase-document relevance in Eqs. (4) and (5), and then optimizes Eqs. (6) and (7) without the graph-based regularization $\mathcal{L}$; (2) **CDA-NMF**: It uses phrase and document embeddings generated by NMF and cosine similarity function to measure phrase-document relevance, and optimizes Eqs. (6) and (7) without $\mathcal{L}$; (3) **CDA-AlterMea**: It uses summation of relevance scores as a commonality measure and differences between relevance scores as a distinction measure; and (4) **CDA-TwoStep**: It first learns relevance score based on Eq. (3.3) and then optimizes Eqs. (6) and (7) without $\mathcal{L}$.

Evaluation Metrics: We use F1 score computed from Precision and Recall to evaluate the performance. Given a document pair, we denote the set of system-identified common terms as $\mathcal{I}$ and the set of gold standard common terms (*i.e.*, words and phrases which are judged as good or perfect) as $\mathcal{G}$. Precision (P) is calculated by $\text{P} = |\mathcal{I} \cap \mathcal{G}|/|\mathcal{I}|$ and Recall (R) is calculated by $\text{R} = |\mathcal{I} \cap \mathcal{G}|/|\mathcal{G}|$. For each document in the pair, we compute above metrics for distinct terms in a similar manner. The reported numbers are averaged over the evaluation set. For parameter study in validation set, we use the same metrics to evaluate the performance.

4.3 Experiments and Performance Study

1. Comparing CDA with the other methods. Tables 4 and 3 summarize the comparison results on the three evaluation sets. Overall, CDA outperforms others on all metrics on all evaluation sets in terms of finding commonalities, and achieves superior Precision and F1 scores on finding distinctions. In particular, CDA obtains a 125% improvement in F1 score and 188% improvement in Recall on the SIGIR dataset compared to the best baseline PatentCom on finding commonalities, and improves F1 on the News dataset by 39.53% compared to PatentCom on finding distinctions.

	Common			Distinct		
Method	**P**	**R**	**F1**	**P**	**R**	**F1**
WordMatch [25]	0.035	0.064	0.045	0.079	0.221	0.112
TopicLabel [27]	0.327	0.449	0.363	0.412	0.851	0.534
PatentCom [43]	0.358	0.481	0.399	0.434	**0.877**	0.554
StringFuzzy	0.180	0.414	0.245	0.376	0.735	0.470
ContextFuzzy	0.166	0.422	0.222	0.317	0.661	0.397
Word2Vec-Clus	0.528	0.213	0.304	0.580	0.347	0.434
NMF-Clus	0.477	0.207	0.289	0.525	0.338	0.411
CDA-AlterMea	0.347	0.613	0.399	0.264	0.194	0.215
CDA-NoGraph	0.600	0.488	0.521	0.838	0.687	0.727
CDA-NMF	0.612	0.654	0.618	0.774	0.699	0.719
CDA-TwoStep	0.642	0.840	0.639	0.831	0.726	0.753
CDA	**0.704**	**0.878**	**0.757**	**0.871**	0.723	**0.773**

Table 3: Performance comparisons on News dataset in terms of Precision, Recall and F1 score.

PatentCom suffers from low recall on commonalities since it finds common terms simply by term overlap without considering semantic common words/phrases (same as WordMatch and TopicLabel). Although its recall on distinctions is high, it has low precision, since it returns a large number of distinct terms without filtering those overly specific ones. Superior performance of CDA validates the effectiveness of our salient phrase generation (vs. WordMatch and TopicLabel) and of the proposed hypotheses on modeling semantic commonality and document-phrase relevance. Both StringFuzzy and ContextFuzzy can find semantic common phrases but they suffer from low precision and instable recall due to its sensitivity to the cut-off threshold. A one-fit-all threshold is not guaranteed to work well for different document pairs or for different domains. Clustering-based methods (*e.g.*, Word2Vec-Clus) yields low-recall results, as it is hard to decide the appropriate cluster granularity and thus many good phrases (which are not close to the centroid) are missed. It is worth mentioning that CDA performs more stably since it leverages the graph-based semantic relevance and the constraints in the optimization, which can control the size of the output sets automatically with adaptive thresholds.

2. Comparing CDA with its variants. Comparing with CDA-NoGraph and CDA-NMF, CDA gains performance from propagating semantic relevance on graphs. Superior performance over CDA-TwoStep further shows the benefit from integrating relevance propagation with phrase selection in a mutually enhancing way. CDA dramatically outperforms CDA-AlterMea on all metrics, which demonstrates the effectiveness of the proposed commonality and distinction measures (see Sec. 3.2).

3. More related pairs versus less related pairs. Fig. 8 compares the methods on pairs of highly similar (more related) documents and lowly similar (less related) documents, respectively. CDA outperforms other methods in terms of Precision and Recall on both kinds of document pairs. As there exist more semantic commonalities and subtle differences between a pair of highly similar documents, CDA gains larger improvement by optimizing the proposed measures and learning semantic relevance. The superior Recall of CDA over CDA-NoGraph (a 23% improvement on the less related pairs) mainly comes from graph-based relevance propagation.

[5] https://code.google.com/archive/p/word2vec/

[6] http://www.cs.cmu.edu/~dpelleg/kmeans.html

Method	SIGIR (common)			SIGIR (distinct)			KDD (common)			KDD (distinct)		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
WordMatch [25]	0.093	0.052	0.062	0.063	0.132	0.081	0.016	0.005	0.007	0.035	0.095	0.049
TopicLabel [27]	0.020	0.016	0.018	0.158	0.427	0.226	0.010	0.003	0.004	0.100	0.248	0.137
PatentCom [43]	0.493	0.292	0.346	0.285	**0.696**	0.413	0.563	0.379	0.423	0.291	**0.732**	0.420
StringFuzzy	0.181	0.815	0.283	0.261	0.494	0.329	0.220	0.802	0.320	0.299	0.631	0.391
ContextFuzzy	0.128	0.839	0.210	0.259	0.335	0.281	0.171	0.796	0.248	0.301	0.485	0.338
Word2Vec-Clus	0.719	0.154	0.255	0.429	0.328	0.384	0.802	0.212	0.335	0.431	0.369	0.398
NMF-Clus	0.687	0.149	0.245	0.408	0.342	0.372	0.787	0.207	0.328	0.415	0.358	0.384
CDA-AlterMea	0.106	0.360	0.157	0.100	0.037	0.049	0.088	0.324	0.131	0.128	0.029	0.044
CDA-NoGraph	0.628	0.637	0.630	0.670	0.529	0.562	0.799	0.705	0.716	0.756	0.645	0.676
CDA-NMF	0.647	0.651	0.649	0.682	0.537	0.601	0.768	0.689	0.726	0.755	0.677	0.714
CDA-TwoStep	0.711	0.818	0.749	**0.708**	0.550	0.597	0.721	0.825	0.749	0.763	0.720	0.726
CDA	**0.752**	**0.843**	**0.778**	0.704	0.596	**0.644**	**0.807**	**0.834**	**0.813**	**0.788**	0.711	**0.733**

Table 4: Performance comparisons on Academia dataset in terms of Precision, Recall and F1 score.

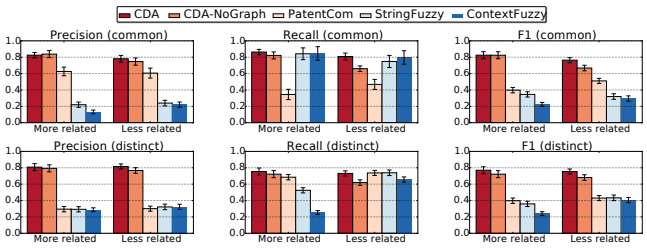
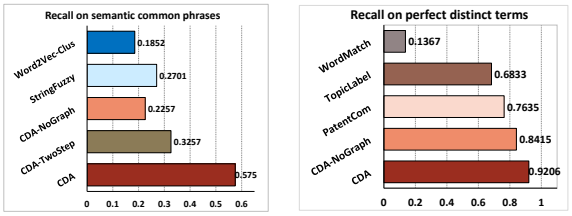


Figure 8: Performance study on “more related" and “less related" document pairs on the Academia dataset.

Distinctions of [14] **Keywords**: search, Web graph, link structure, *PageRank*, search in context, personalized search	**Distinctions of** [9] **Keywords**: web search, *PageRank*
hub, partial, skeleton, pages, personalized, section, computation, preference	query, rank, htm, sensitive, ranking, urls, search, topic, context, regional
The Hubs Theorem allows basis vectors to be encoded as partial vectors and a hubs skeleton. Our approach enables incremental computation, so that the construction of personalized views from partial vectors is practical at query time.	Finally, we compute the query-sensitive importance score of each of these retrieved URLs as follows. In Section 4.3, the topic-sensitive ranking vectors were chosen using the topics most strongly associated with the query term contexts.
personalized web search, user-specified⊕web pages, dynamic programming, incremental computation, theoretical results	topic-sensitive PageRank, query topic, context-specific importance score, query context, topic-sensitive⊕ranking vector

Table 5: Distinctions for [14] and [9] generated by WordMatch [25] (top), Discriminative Sentence Selection [39] (middle), and CDA (bottom).



(a) Semantic Commonality (b) Perfect Distinction

Figure 9: Case studies on phrase semantic commonality and perfect distinction on the News dataset.

4.4 Case Study

1. Comparing with word-based and sentence-based summarization. Table 5 shows the comparative analysis results between papers [14] and [9] generated by word-based method [25], sentence-based method [39] (which gives top-2 sentences), and our phrase-based CDA approach. We do not include commonality results since sentence-based summarization techniques only provide distinction results [39] for each document. We found that CDA provides sufficiently cohesive and readable results compared with word-based methods, and it keeps the overall summary concise, as compared to sentence-based methods. Furthermore, the results also show that author-generated keywords are often not able to highlight the distinctions when compared to other papers.

2. Testing on semantic commonality. To study the performance on finding semantic common phrases (Fig. 3), we compare our method with the baselines which can also find such phrases. In Fig. 9(a). CDA achieved significant improvement in recall since it leverages the bipartite graph to derive semantic relevance (versus CDA-NoGraph, StringFuzzy, Word2Vec-Clus), and integrates the relevance score propagation with phrase selection to reinforce the learning of semantic relevance (versus CDA-TwoStep). Compared with StringFuzzy, our method does not require a unified cut-off threshold and thus is more robust across different document pairs.

3. Testing on perfect distinction. We consider *overly general* terms as positive in previous evaluations (*i.e.*, “good" labels are given to overly general terms in Sec. 4.2). Next, we further test our method particularly on finding “perfect" distinct terms (by assigning “bad" label to those overly general terms). In Fig. 9(b), CDA achieved superior performance (over 90% on recall) compared with other methods. This is because that (1) phrase is not only informative and concise enough for user to understand, but also general enough to highlight the distinctions (versus WordMatch); and (2) our relevance score learning can balance phrase generality and discrimination well so as to be not dominated by overly general terms. (versus CDA-NoGraph, PatentCom, TopicLabel).

4. Comparing two document sets. Fig. 10 presents CDA output for comparing document sets on News dataset (200 documents were sampled for each date). Common phrases show the connections between things happened in two different dates while distinction phrases highlight the unique things happened in each date. In particular, distinction results demonstrate that our method can capture pairwise distinction property well by generating different distinct phrases for the same news set when comparing with different news sets. The comparison results provide a good overview on the event evolution of 2011 Japan Tsunami.

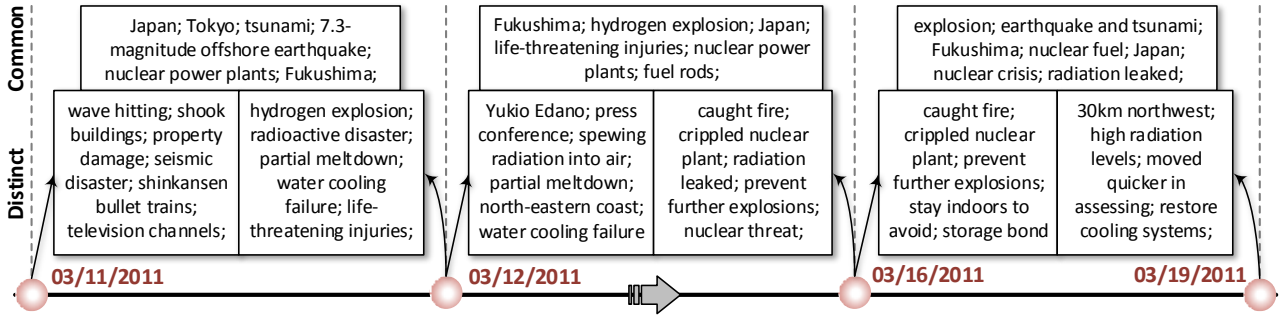


Figure 10: Compare document sets on News dataset. We use news articles published at four different dates.

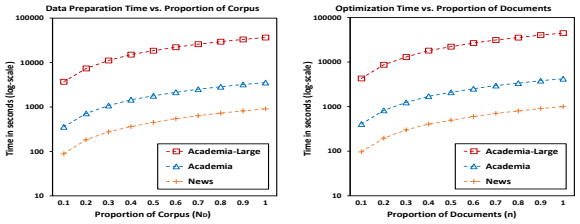
4.5 Scalability

To evaluate the time complexity of PhraseCom, we compute the runtime for subsets of the original dataset (created by randomly sampling with different ratios). In addition to the Academia and News datasets, we also test on a collection of 2.58M academic publications (denoted as Acad-Large). Table 6 shows the dataset statistics and PhraseCom's runtime on 10,000 document pairs[7].

Dataset	File Size	#Words	#Docs	Time
News	231MB	44M	67,809	0.53(hrs)
Academia	937MB	158M	205,484	1.64(hrs)
Acad-Large	11.7GB	1.98B	2.58M	20.37(hrs)

Table 6: Runtime of PhraseCom on 10,000 document pairs.

As discussed in Sec. 3.4, PhraseCom can be decomposed into two main separate steps, *i.e.*, data preparation, and comparative selection optimization (for each document pair). Figs. 11(a) and 11(b) demonstrate the runtime trends for these two steps on the three datasets, where time is displayed on a log-scale for ease of interpretation. In both cases, runtime of our method seems to scale linearly as we increase the size of the corpus. This verifies the analysis in Sec. 3.4. Comparison with other systems (*e.g.*, Discriminative Sentence Selection [39]) are not conducted since these systems are implemented by different programming languages.



(a) Data preparation time (b) Optimization time vs. %docs

Figure 11: Runtime analysis of PhraseCom on the three datasets.

5. RELATED WORK

There have been many attempts on performing comparative analysis on text data. Previous work can be categorized in terms of sources of comparison (*e.g.*, single or multiple corpora), targets of comparison (*e.g.*, between topics, individual documents or document sets), aspects to compare (*e.g.*, commonality, distinction or both), and representation forms of results (*e.g.*, words, sentences).

Multi-document summarization [34, 7, 21, 4, 31, 6, 5] aims to generate a compressed summary to cover the *consensus* of information among the original documents. It is different from *commonality* discovery as it focuses on providing comprehensive view of the corpus (union instead of intersection).

Comparative document summarization [43, 24, 13, 39, 47, 12, 38, 16] focuses on the *distinction* aspect—it summarizes the differences between two document sets by extracting the discriminative sentences from each set. Existing work formalizes the problem of discriminative sentence selection into different forms, including integer linear programming [13], multivariate model estimation [39], and group-related centrality estimation [25]. Going beyond selecting discriminative sentences from a document set, our proposed CDA task aims to select quality and concise phrases to highlight not only differences but also commonalities between two documents.

In particular, our work is related to [25] since both try to derive common and distinct terms for a pair of related documents, but their work focuses on finding topic-related terms and selecting sentences based on such terms. They assume terms appearing in both documents as the common ones, and treat the remaining terms as distinct ones. As shown in our experiments, this method (labeled as WordMatch) suffers from low recall in finding common terms and low precision in finding distinct terms since it ignores semantic common phrases and pairwise distinct phrases. Similarly, Zhang *et al.* [43] consider both common and distinct aspects in generating comparative summary for patent documents. They derive a term co-occurrence tree which can be used to extract sentences for summarization. However, they use all shared noun phrases between two documents as common terms, and apply feature selection techniques to find distinct terms. This method (see PatentCom in Sec. 4.3) demonstrates poor performance on finding common terms due to the ignorance of semantic common phrases; although this method performs well in terms of the recall for distinct phrases, it achieves low precision, since it fails to model phrase generality and produce many overly-specific phrases.

Another line of related work, referred to as comparative text mining [42], focuses on modeling latent comparison aspects and discovering clusters of common and distinct words for each aspect. They adopt topic model [3, 42, 23] and matrix factorization [17] to present the common and distinct information by multinomial distributions of words. While latent comparison aspects can help enrich the comparison results, these methods still adopt bag-of-words representation which is often criticized for its unsatisfying readability [27]. Furthermore, it is difficult to apply statistical topic modeling in comparative document analysis as the data statistic between two documents is insufficient, in particular when the documents are about emergent topics (*e.g.*, News). Finally, our work is also related to comparing reviews in opinion mining [40, 36, 35, 29, 18, 15] and contrastive summarization for entities [20]—they also aim to find similarities and differences between two objects. However, these works are restricted to sentiment analysis.

[7]The execution time experiments were all conducted on a machine with 4 cores of Intel i5-2500 CPU@3.30GHz, following the setup introduced in Secs. 4.1 and 4.2.

6. CONCLUSION AND FUTURE WORK

In this paper, we study the problem of phrase-based comparative summarization for a document pair, called comparative document analysis (CDA), and propose a general graph-based approach to model semantic commonality and pairwise distinction for phrases. We cast the phrase selection problems into joint optimization problems based on the proposed novel measures. Experiment results demonstrate the effectiveness and robustness of the proposed method on text corpora of different domains. Interesting future work includes extending CDA to consider different comparison aspects [42, 17] and to exploit the hierarchical semantic relations between phrases. CDA is general and can be applied as a primitive step for sentence-based comparative summarization [13, 39]. It can potentially benefit many other applications such as content recommendation [33, 41], entity extraction [32] and relevance feedback [26].

7. ACKNOWLEDGMENTS

This work was partially done when the first author was an intern in Microsoft Research, Redmond. Research was sponsored in part by the U.S. Army Research Lab. under Cooperative Agreement No. W911NF-09-2-0053 (NSCTA), National Science Foundation IIS-1017362, IIS-1320617, and IIS-1354329, HDTRA1-10-1-0120, and grant 1U54GM114838 awarded by NIGMS through funds provided by the trans-NIH Big Data to Knowledge (BD2K) initiative (www.bd2k.nih.gov). The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

8. REFERENCES

[1] S. Bedathur, K. Berberich, J. Dittrich, N. Mamoulis, and G. Weikum. Interesting-phrase mining for ad-hoc text analytics. *VLDB*, 2010.

[2] A. Bordes, J. Weston, R. Collobert, and Y. Bengio. Learning structured embeddings of knowledge bases. In *AAAI*, 2011.

[3] C. Chen, W. Buntine, N. Ding, L. Xie, and L. Du. Differential topic models. *TPAMI*, 37(2):230–242, 2015.

[4] J. M. Conroy and D. P. O'leary. Text summarization via hidden markov models. In *SIGIR*, 2001.

[5] J. Goldstein, V. Mittal, J. Carbonell, and M. Kantrowitz. Multi-document summarization by sentence extraction. In *NAACL-ANLP Workshop on Automatic summarization*, 2000.

[6] Y. Gong and X. Liu. Generic text summarization using relevance measure and latent semantic analysis. In *SIGIR*, 2001.

[7] A. Haghighi and L. Vanderwende. Exploring content models for multi-document summarization. In *NAACL*, 2009.

[8] Z. S. Harris. Distributional structure. *Word*, 1954.

[9] T. H. Haveliwala. Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search. *TKDE*, 15(4):784–796, 2003.

[10] J. He, H. Tong, Q. Mei, and B. Szymanski. Gender: A generic diversified ranking algorithm. In *NIPS*, 2012.

[11] G. E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

[12] X. Huang, X. Wan, and J. Xiao. Comparative news summarization using linear programming. In *ACL*, 2011.

[13] X. Huang, X. Wan, and J. Xiao. Comparative news summarization using concept-based optimization. *Knowledge and information systems*, 38(3):691–716, 2014.

[14] G. Jeh and J. Widom. Scaling personalized web search. In *WWW*, 2003.

[15] N. Jindal and B. Liu. Identifying comparative sentences in text documents. In *SIGIR*, 2006.

[16] N. Jindal and B. Liu. Mining comparative sentences and relations. In *AAAI*, 2006.

[17] H. Kim, J. Choo, J. Kim, C. K. Reddy, and H. Park. Simultaneous discovery of common and discriminative topics via joint nonnegative matrix factorization. In *KDD*, 2015.

[18] H. D. Kim and C. Zhai. Generating comparative summaries of contradictory opinions in text. In *CIKM*, 2009.

[19] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *NIPS*, 2001.

[20] K. Lerman and R. McDonald. Contrastive summarization: an experiment with consumer reviews. In *NAACL*, 2009.

[21] C.-Y. Lin and E. Hovy. From single to multi-document summarization: A prototype system and its evaluation. In *ACL*, 2002.

[22] J. Liu, J. Shang, C. Wang, X. Ren, and J. Han. Mining quality phrases from massive text corpora. In *SIGMOD*, 2015.

[23] Y. Lu and C. Zhai. Opinion integration through semi-supervised topic modeling. In *WWW*, 2008.

[24] A. S. Maiya. A framework for comparing groups of documents. *EMNLP*, 2015.

[25] I. Mani and E. Bloedorn. Multi-document summarization by graph search and matching. *AAAI*, 1997.

[26] C. D. Manning, P. Raghavan, H. Schütze, et al. *Introduction to information retrieval*. Cambridge university press, 2008.

[27] Q. Mei, X. Shen, and C. Zhai. Automatic labeling of multinomial topic models. In *SIGKDD*, 2007.

[28] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013.

[29] M. J. Paul, C. Zhai, and R. Girju. Summarizing contrastive viewpoints in opinionated text. In *EMNLP*, 2010.

[30] D. Pelleg, A. W. Moore, et al. X-means: Extending k-means with efficient estimation of the number of clusters. In *ICML*, 2000.

[31] D. R. Radev, H. Jing, and M. Budzikowska. Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation, and user studies. In *NAACL*, 2000.

[32] X. Ren, W. He, M. Qu, H. Ji, C. R. Voss, and J. Han. Label noise reduction in entity typing by heterogeneous partial-label embedding. In *KDD*, 2016.

[33] X. Ren, J. Liu, X. Yu, U. Khandelwal, Q. Gu, L. Wang, and J. Han. ClusCite: effective citation recommendation by information network-based clustering. In *KDD*, 2014.

[34] C. Shen and T. Li. Multi-document summarization via the minimum dominating set. In *COLING*, 2010.

[35] R. Sipos and T. Joachims. Generating comparative summaries from reviews. In *CIKM*, 2013.

[36] M. Tkachenko and H. W. Lauw. Generative modeling of entity comparisons in text. In *CIKM*, 2014.

[37] P. Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *JOTA*, 109(3):475–494, 2001.

[38] X. Wan, H. Jia, S. Huang, and J. Xiao. Summarizing the differences in multilingual news. In *SIGIR*, 2011.

[39] D. Wang, S. Zhu, T. Li, and Y. Gong. Comparative document summarization via discriminative sentence selection. *TKDD*, 6(3):12, 2013.

[40] S. Wang, Z. Chen, and B. Liu. Mining aspect-specific opinion using a holistic lifelong topic model. In *WWW*, 2016.

[41] X. Yu, X. Ren, Y. Sun, Q. Gu, B. Sturt, U. Khandelwal, B. Norick, and J. Han. Personalized entity recommendation: a heterogeneous information network approach. In *WSDM*, 2014.

[42] C. Zhai, A. Velivelli, and B. Yu. A cross-collection mixture model for comparative text mining. In *SIGKDD*, 2004.

[43] L. Zhang, L. Li, C. Shen, and T. Li. Patentcom: A comparative view of patent document retrieval. *SDM*, 2015.

[44] Z. Zhang. A comparative evaluation of term recognition algorithms. In *LERC*, 2008.

[45] D. Zhou, J. Weston, A. Gretton, and O. Bousquet. Ranking on data manifolds. *NIPS*, 2004.

[46] X. Zhu, J. Lafferty, and R. Rosenfeld. *Semi-supervised learning with graphs*. Carnegie Mellon University, 2005.

[47] F. Zhuang, P. Luo, Z. Shen, Q. He, Y. Xiong, Z. Shi, and H. Xiong. Mining distinction and commonality across multiple domains using generative model for text classification. *TKDE*, 24(11):2025–2039, 2012.