Published in final edited form as:
*KDD*. 2015 August ; 2015: 1215–1224. doi:10.1145/2783258.2783374.

# Incorporating World Knowledge to Document Clustering via Heterogeneous Information Networks

**Chenguang Wang**[†], **Yangqiu Song**[‡], **Ahmed El-Kishky**[‡], **Dan Roth**[‡], **Ming Zhang**[†], and **Jiawei Han**[‡]

Chenguang Wang: wangchenguang@pku.edu.cn; Yangqiu Song: yqsong@illinois.edu; Ahmed El-Kishky: elkishk2@illinois.edu; Dan Roth: danr@illinois.edu; Ming Zhang: mzhang_cs@pku.edu.cn; Jiawei Han: hanj@illinois.edu

[†]School of EECS, Peking University

[‡]Department of Computer Science, University of Illinois at Urbana-Champaign

## Abstract

One of the key obstacles in making learning protocols realistic in applications is the need to supervise them, a costly process that often requires hiring domain experts. We consider the framework to use the world knowledge as indirect supervision. World knowledge is general-purpose knowledge, which is not designed for any specific domain. Then the key challenges are how to adapt the world knowledge to domains and how to represent it for learning. In this paper, we provide an example of using world knowledge for domain dependent document clustering. We provide three ways to specify the world knowledge to domains by resolving the ambiguity of the entities and their types, and represent the data with world knowledge as a heterogeneous information network. Then we propose a clustering algorithm that can cluster multiple types and incorporate the sub-type information as constraints. In the experiments, we use two existing knowledge bases as our sources of world knowledge. One is Freebase, which is collaboratively collected knowledge about entities and their organizations. The other is YAGO2, a knowledge base automatically extracted from Wikipedia and maps knowledge to the linguistic knowledge base, Word-Net. Experimental results on two text benchmark datasets (20newsgroups and RCV1) show that incorporating world knowledge as indirect supervision can significantly outperform the state-of-the-art clustering algorithms as well as clustering algorithms enhanced with world knowledge features.

### Keywords

World Knowledge; Heterogeneous Information Network; Document Clustering; Knowledge Base; Knowledge Graph

## 1. Introduction

Machine learning algorithms have become pervasive in multiple domains, impacting a wide variety of applications. Nonetheless, a key obstacle in making learning protocols realistic in applications is the need to supervise them, a costly process that often requires hiring domain experts. In the past decades, machine learning community has elaborated to reduce the labeling work done by human for supervised machine learning algorithms or to improve unsupervised learning with only minimum supervision. For example, semi-supervised learning [6] is proposed to use only partially labeled data and a lot of unlabeled data to perform learning with the hope that it can perform as good as fully supervised learning. Transfer learning [28] uses the labeled data from other relevant domains to help the learning task in the target domain. However, there are still many cases that neither semi-supervised learning nor transfer learning can help. For example, in the era of big data, we can have a lot textual information from different Web sites, e.g., blogs, forums, mailing lists. It is impossible to ask human to annotate all the required tasks. It is also difficult to find relevant labeled domains. Recognizing that some domains can be very specific and really need the domain experts to perform annotation, e.g., the medical domain publication classification. Therefore, we should consider a more general approach to further reducing the labeling cost for learning tasks in diverse domains.

Fortunately, with the proliferation of general-purpose knowledge bases (or knowledge graphs), e.g., Cyc project [18], Wikipedia, Freebase [5], KnowItAll [9], TextRunner [2], WikiTaxonomy [29], Probase [40], DBpedia [1], YAGO [35], NELL [26] and Knowledge Vault [8], we have an abundance of available world knowledge. We call these knowledge bases world knowledge [11], because they are universal knowledge that are either collaboratively annotated by human labelers or automatically extracted from big data. When world knowledge is annotated or extracted, it is not collected for any specific domain. However, because we believe the facts in world knowledge bases are very useful and of high quality, we propose using them as supervision for many machine learning problems. People have found it useful to use world knowledge as distant supervision for entity and relation extraction and embedding [25, 39, 41]. This is a direct use of the facts in world knowledge bases, where the entities in the knowledge bases are matched in the context regardless the ambiguity. A more interesting question is *can we use the world knowledge to indirectly "supervise" more machine learning algorithms or applications?* For example, if we can use world knowledge as indirect supervision, then we can extend the knowledge about entities and relations to more generic text analytics problems, e.g., categorization and information retrieval.

Thus, we consider a general machine learning framework that can incorporate world knowledge into machine learning algorithms. As mentioned, world knowledge is not designed for any specific domain. For example, when we want to cluster the documents about entertainment or sports, then the world knowledge about names of celebrities and athletes may help while the terms used in science and technology may not be very useful. Thus, a key issue is how we should adapt world knowledge to the domain specific tasks. Another problem is when we have the world knowledge, how we can represent it for the domain dependent tasks. For example, because most of the knowledge bases use a linked network to organize the knowledge, to adapt the world knowledge to domains, we should consider how to use the linked data. Although traditional machine learning algorithms using world knowledge just treat world knowledge as “flat” features in addition to the original text data [11, 22], the structure of the knowledge provides rich information about the connections of entities and relations. Therefore, we should also carefully consider the best way to represent the world knowledge for machine learning algorithms.

In this paper, we illustrate a framework of machine learning with world knowledge using a document clustering problem. We select two knowledge bases, i.e., Freebase, YAGO2, as the sources of world knowledge. Freebase [5] is a collaboratively collected knowledge base about entities and their organizations. YAGO2 [35] is a knowledge base automatically extracted from Wikipedia and maps the knowledge to the linguistic knowledge base, WordNet [10]. To adapt the world knowledge to domain specific tasks, we first use semantic parsing to ground any text to the knowledge bases [4]. We then apply frequency, document frequency, and conceptualization [32] based semantic filters to resolve the ambiguity problem when adapting world knowledge to the domain tasks. After that, we have the documents as well as the extracted entities and their relations. Since the knowledge bases provide the entity types, the resulting data naturally form a heterogeneous information network (HIN) [13]. We show an example of such HIN in Figure 1. The specified world knowledge, such as named entities (“Bush”, “Obama”) and their types (*Person*), as well as the documents and the words form the HIN. We then formulate the document clustering problem as an HIN partitioning problem, and provide a new algorithm to better perform clustering by incorporating the rich structural information as constraints in the HIN. For example, the HIN builds a link (a must-link constraint) between “Obama” of sub-type *Politician* in one document and “Bush” of sub-type *Politician* in another document. Such link and type information could be very useful if the target clustering domain is “Politics.”

The main contributions of this work are highlighted as follows:

- We study a novel problem of supervising machine learning algorithms with world knowledge.
- We propose to use semantic parsing and semantic filtering to specify world knowledge to the domain dependent documents, and develop a new constrained HIN clustering algorithm to make better use of the structural information from the world knowledge for document clustering task.
- We conduct experiments on two benchmark datasets (20newsgroups and RCV1) to evaluate the clustering algorithm using HIN, compared with the state-of-the-art document clustering algorithms and clustering with “flat” world knowledge features. We show that our approach can be 13.3% better than the semi-supervised clustering algorithm incorporating 250K constraints which are generated by ground-truth labels.

## 2. To Learn with World Knowledge

In this section, we discuss how we enable world knowledge to indirectly “supervise” machines, instead of just using world knowledge as additional features. In general, performing machine learning with world knowledge, we should follow four steps: (1) Knowledge acquisition. (2) Domain adaptation. (3) Data and knowledge representation. (4) Learning. Since we assume the world knowledge is given, we skip step one in this study. Then given the world knowledge, we should consider adapting it to specific domains. Since the knowledge can be ambiguous without context, we should consider using domain dependent data to find the best knowledge to use. For example, when a text mentions "apple," it can refer to a company or a fruit. In the knowledge base, we have both. Therefore, we should choose the right one to use. Then given the filtered knowledge we have as well as the domain dependent data, we use a better representation which considers the structure information of the linked knowledge rather than just considering the knowledge as flat features. After we have the representation, we can design a learning algorithm for domain dependent task.

The above four steps are general, which means they may apply to many applications. In this section, we demonstrate how to select the right knowledge to use and to represent this knowledge for the task of document clustering. Then in the next section, we will introduce the learning algorithm to perform better document clustering given the representation.

### 2.1 World Knowledge Specification

In this subsection, we propose a world knowledge specification approach to generate specified world knowledge given a set of domain dependent documents. We first use semantic parsing to ground any text to the knowledge base, then provide three semantic filtering approaches to avoid ambiguity of the extracted information.

**2.1.1 Semantic Parsing—**Semantic parsing is the task of mapping a piece of natural language text to a formal meaning representation [27]. This can support question answering by querying a knowledge base [17]. To our best knowledge, most previous semantic parsing algorithms or tools developed are for small scale problems but with complicated logical forms, until Berant et al. [4] develop a system that can handle very large scale knowledge bases such as Freebase. They use the developed system to solve question answering problem with Freebase. In their work, they formulate their problem to match answers to the questions, which is a supervised learning process. Similar to them, we are also working with very large scale world knowledge bases, but unlike them, we do not match question and answers. Our task is to ground any text to the knowledge base entities and their relationships in the prescribed logical form. Therefore, our problem is a fully unsupervised problem.

We first introduce the problem formulation and then introduce how we perform unsupervised semantic parsing. Let $\mathscr{E}$ be a set of entities and $\mathscr{R}$ be a set of relations in the knowledge base. Then the knowledge base $\mathscr{K}$ consists of triplets in the form of $(e_1, r, e_2)$, where $e_1, e_2 \in \mathscr{E}$ and $r \in \mathscr{R}$. We follow [4] to use a simple version of Lambda Dependency-Based Compositional Semantics ($\lambda$-DCS) [21] as the logic language. From each sentence in the document, we can parse four possible $\lambda$-DCS logic forms [4]: (1) Unary: an entity e is a unary logic form (e.g., *Obama*); (2) Binary: a relation $r$ is a binary logic form (e.g., *PresidentofCountry*); (3) Join: $r.e$ is a unary logic form, denoting a join, where $r$ is a binary and $e$ is a unary (e.g., *PresidentofCountry.Obama*); (4) Intersection: $e_1 \bigcap e_2$ $(e_1, e_2 \in \mathscr{E})$

denotes set intersection, where $e_1$ and $e_2$ are both unaries (e.g., *Location.Olympics* $\bigcap$ *PresidentofCountry.Obama*).

In simpler terms, semantic parsing can be understood as the following process. First, given a piece of text "Obama is the president of United States of America," it maps the entities as well as the relation phrases in the text to knowledge base. So "Obama" and "United States of America" are mapped to knowledge base, resulting in two unary logic forms *People.BarackObama* and *Country.USA*, where *People* and *Country* are the type information in Freebase. The relation phrase "president" is mapped to a binary logic form *PresidentofCountry*. Notice that, the mapping process skips the words "is" and "of." The mapping dictionary is constructed by aligning a large text corpus to the knowledge base. A phrase and a knowledge base entity or relation can be aligned if they co-occur with many of the same entities. We select two knowledge bases, i.e., Freebase and YAGO2. For Freebase, we just use the mapping already existing in the released tool shown in [4]. For YAGO2, we follow [4] and download a subset of ClueWeb09[1] to find the new mapping for YAGO2 entities and relations. Second, it uses some rules (i.e., grammar) to combine the basic logic forms to generate the restricted four logic forms above, and rank the results. For the example shown in this paragraph, *People.BarackObama* $\wedge$ *President.USA* is generated to represent its semantic meaning. Notice that, *President.USA* is generated by joining the unary *Country.USA* with the binary *PresidentofCountry*.

When there are more than one candidate semantic meanings for a sentence, in [4], they learn the ranks based on the annotated question-answer pairs. For our task, this annotation is not available. Therefore, instead of ranking or enumerating all the possible logic forms (which is found to be not feasible in limited time), we constrain the entities to be the maximum length spanning phrases recognized by a state-of-the-art named entity recognition tool [30]. We then perform the two steps introduced above by using the maximum length spanning noun phrase as entities, and use the phrase between them in the text as relation phrase. We propose to use the following three semantic filtering methods to resolve the ambiguation problem.

### 2.1.2 Semantic Filtering—

For each sentence in the given document, the output of semantic parsing is a set of logic forms that represent the semantic meaning. However, the extracted entities can be ambiguous. For example, "apple" may be associated with type *Company* or *Fruit*. Therefore, we should filter out the noisy entities and their types to ensure that the knowledge we have is good enough as indirect supervision for document clustering. We assume that in the domain specific tasks, given the context, the entities seldom have multiple meanings. Thus, we propose the following three approaches to select the best knowledge to use for further learning process.

**Frequency based semantic filter (FBSF):** We use the frequency of a type for an entity appearing in a document as the criterion to decide whether the entity should be extracted for the domain specific task in a sentence. Here we assume the most frequent type of an entity from all the sentences of the document is the correct semantic meaning.

[1] http://www.lemurproject.org/clueweb09.php/

**Document frequency based semantic filter (DFBSF):** Similar to the frequency based method, we use the document frequency (DF) of a type of an entity as the criterion to find the most likely semantic meaning. Here we assume that if an entity appears in multiple documents with the same type, then the type should be the correct semantic meaning.

**Conceptualization based semantic filter (CBSF):** Motivated by the approaches of conceptualization [32, 33] and entity disambiguation [20], we represent each entity with a feature vector of entity types, and use standard Kmeans to cluster the entities. Then in each cluster, we use the intersection operation to find the most likely entity type for the entities in the cluster. In this case, different entities can be used to disambiguate each other. Here we assume that the type that can best fit the context is the correct semantic meaning.

## 2.2 World Knowledge Representation

The output of semantic parsing and semantic filtering is then the document associated with the entities, which are further associated with the types (or concepts, categories, the names can be different for different knowledge bases and relations). For example, in Freebase, we select the top level named entity categories (i.e., domains) as the types, e.g., *Person*, *Location*, and *Organization*. In addition to the named entities, we also regard the document and word as two types. Then we use an HIN to represent the data we get after semantic parsing and semantic filtering.

Definition 1. A **heterogeneous information network** (HIN) is a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with an entity type mapping $\varphi$: $\mathcal{V} \to \mathcal{A}$ and a relation type mapping $\psi$: $\mathcal{E} \to \mathcal{R}$, where $\mathcal{V}$ denotes the entity set and $\mathcal{E}$ denotes the link set, $\mathcal{A}$ denotes the entity type set and $\mathcal{R}$ denotes the relation type set, and the number of entity types $|\mathcal{A}| > 1$ or the number of relation types $|\mathcal{R}| > 1$.

The network schema provides a high-level description of a given heterogeneous information network.

Definition 2. Given an HIN $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with the entity type mapping $\varphi$: $\mathcal{V} \to \mathcal{A}$ and the relation type mapping $\psi$: $\mathcal{E} \to \mathcal{R}$, the **network schema** for network $G$, denoted as $\mathcal{T}_{\mathcal{G}} = (\mathcal{A}, \mathcal{R})$, is a graph with nodes as entity types from $\mathcal{A}$ and edges as relation types from $\mathcal{R}$.

Then for our world knowledge dependent network, we use the network schema shown in Figure 2 to represent the data. The network contains multiple entity types: **document** $\mathcal{D}$, **word** $\mathcal{W}$, **named entities** $\{\mathcal{E}^{\mathrm{I}}\}_{I=1}^{T}$, and a few relation types connecting the entity types. Notice that, we use "entity type" to represent the node type in HIN, as Definition 1 showed. We use "named entity type" to represent the type of the name mentioned in text (widely used in NLP community), e.g., person, location, and organization names. The entities in HIN do not have to be named entities, e.g., the categories of animals or diseases. We denote the document set as $\mathcal{D} = \{d_1, d_2, \ldots, d_M\}$, where $M$ is the size of $\mathcal{D}$, the word set as $\mathcal{W} = \{w_1, w_2, \ldots, w_N\}$, where $N$ is the size of $\mathcal{W}$, and the entity set as $\mathcal{E}^t = \{e_1^t, e_2^t, \ldots, e_{V_t}^t\}$, where $V_t$ is the size of $\mathcal{E}^t$. We have $t = 1, \ldots, T$ where $T$ is the total number of named entity types we find in the knowledge base. Note that if there are no named entities, then the network reduces to a bipartite graph containing only documents and words.

## 3. Document Clustering with World Knowledge

In this subsection, we present our clustering algorithm using HIN, constructed from domain dependent documents and the world knowledge. Given the HIN, it is natural to perform HIN partitioning to obtain the document clusters. In addition to the HIN itself, let us revisit the structural information in a typical world knowledge base, e.g., Freebase. In the world knowledge base, the named entities are often organized in a hierarchy of categories. Although there are additional category information for each entity, we only use the top level named entity types as the entity types in HIN. For example, “Barack Obama” is a person, where person is the top level category. In addition, he is the president of the “United States,” a politician, a celebrity, etc.. Another example is that “Google” is a software company, plus it has a CEO. This shows that the entities can have some attributes. We choose to use top level entity types for the HIN schema since then we will have a relatively dense graph for each pairwise nodes in the network schema. The fine-grained named entity sub-types or the attributes are also very useful to identify the topics or the clusters of the documents. Therefore, in this section, we introduce how we incorporate the fine-grained level of named entity types as constraints in the HIN clustering algorithm.

### 3.1 Constrained Clustering Modeling

To formulate the clustering algorithm for the domain dependent documents, we denote latent label sets of the documents as $\mathscr{L}_d = \{l_{d1}, l_{d2},\ldots l_{dM}\}$. We also denote $\mathscr{L}_w = \{l_{w1}, l_{w2},\ldots,l_{wN}\}$ for words, and $\mathscr{L}_{e^t}=\{l_{e_1^t}, l_{e_2^t}, \ldots, l_{e_{V_t}^t}\}$ for the $t^{th}$ named entities set. In general, we follow the framework of information-theoretic co-clustering (ITCC) [7] and constrained ITCC [31, 38] to formulate our approach. Instead of only performing on the bipartite graph, we need to handle multi-type relational data, as well as more complicated constraints.

The original ITCC uses a variational function to approximate the joint probability of documents and words, which is:

$$q(d_m, w_i)=p(\hat{d}_{k_d}, \hat{w}_{k_w})p(d_m|\hat{d}_{k_d})p(w_i|\hat{w}_{k_w}), \quad (1)$$

where $\hat{d_{k_d}}$ and $\hat{w}_{k_w}$ are cluster indicators to formulate the conditional probability, and $k_d$ and $k_w$ are the corresponding cluster indices. $q(d_m, w_i)$ is used to approximate $p(d_m, w_i)$ by minimizing the Kullback-Leibler (KL) divergence:

$$\begin{aligned} &D_{KL}(p(\mathscr{D},\mathscr{W})||q(\mathscr{D},\mathscr{W})) \\ &=D_{KL}(p(\mathscr{D},\mathscr{W},\hat{\mathscr{D}},\hat{\mathscr{W}})||q(\mathscr{D},\mathscr{W},\hat{\mathscr{D}},\hat{\mathscr{W}})) \\ &=\sum_{k_d}^{K_d}\sum_{d_m:l_{d_m}=k_d}p(d_m)D_{KL}(p(\mathscr{W}|d_m)||p(\mathscr{W}|\hat{d}_{k_d})) \\ &=\sum_{k_w}^{K_w}\sum_{w_i:l_{w_i}=k_w}p(w_i)D_{KL}(p(\mathscr{D}|w_i)||p(\mathscr{D}|\hat{w}_{k_w})), \end{aligned} \quad (2)$$

where $\hat{\mathscr{D}}$ and $\hat{\mathscr{W}}$ are the cluster sets, $p(\mathscr{W}|\hat{d}_{k_d})$ denotes a multinomial distribution based on the probabilities

$$p(\mathscr{W}|\hat{d}_{k_d})=(p(w_1|\hat{d}_{k_d}),\ldots,p(w_N|\hat{d}_{k_d}))^T, p(w_i|\hat{d}_{k_d})=p(w_i|\hat{w}_{k_w})p(\hat{w}_{k_w}|\hat{d}_{k_d})\text{and}, p(w_i|\hat{w}_{k_w})=p(w_i)/p(l_{w_i}=\hat{w}_{k_w}).$$

Symmetrically, we have

$$p(\mathscr{D}|\hat{w}_{k_w})=(p(d_1|\hat{w}_{k_w}),\ldots,p(d_M|\hat{w}_{k_w}))^T, p(d_i|\hat{w}_{k_w})=p(d_i|\hat{d}_{k_d})p(\hat{d}_{k_d}|\hat{w}_{k_w})\text{and}, p(d_i|\hat{d}_{k_d})=p(d_i)/p(l_{d_i}=\hat{d}_{k_d}).$$

Moreover, $p(\hat{w}_{k_w}|\hat{d}_{k_d})$ and $p(\hat{d}_{k_d}|\hat{w}_{k_w})$ are computed based on the joint probability $q(\hat{d}_{k_d}, \hat{w}_{k_w}) = \Sigma_{l_{d_m}=k_d} \Sigma_{l_{w_i}=k_w}\, p(d_m, w_i)$.

Motivated by ITCC, according to the network schema shown in Figure 2, our problem of HIN clustering is formulated as

$$\mathscr{J}_{\text{HINC}}=D_{KL}(p(\mathscr{D},\mathscr{W})\|q(\mathscr{D},\mathscr{W}))+\sum_{t=1}^{T}D_{KL}(p(\mathscr{D},\mathscr{E}^t)\|q(\mathscr{D},\mathscr{E}^t))+\sum_{t=1}^{T}\sum_{s=1}^{T}D_{KL}(p(\mathscr{E}^t,\mathscr{E}^s)\|q(\mathscr{E}^t,\mathscr{E}^s)), \quad (3)$$

where all the probabilities can be defined similar to the document-word bipartite graph. We omit the detailed definitions due to the space limitation. A summary of the notations is shown in Table 1.

To incorporate the side information of the fine-grained named entity sub-types or the attributes as indirect supervision for document clustering, we define the constraints for the named entities we find after semantic parsing. We take the $t^{th}$ entity label set $\mathscr{E}^t$ as an example, and use must-links and cannot-links as the constraints. We denote the must-link set associated with $e_i^t$ as $\mathscr{M}_{e_i^t}$, and the cannot-link set as $\mathscr{C}_{e_i^t}$. For must-links, the cost function is defined as

$$V_{\mathscr{M}}(e^t_{i_1}, e^t_{i_2} \in \mathscr{M}_{e^t_{i_1}})=w_{\mathscr{M}}D_{KL}(p(\mathscr{D}|e^t_{i_1})\|p(\mathscr{D}|e^t_{i_2}))\cdot\mathscr{I}_{l_{e^t_{i_1}}\neq l_{e^t_{i_2}}}, \quad (4)$$

where $w_{\mathscr{M}}$ is the weight for must-links, and $p(\mathscr{D}|e^t_{i_1})$ denotes a multinomial distribution based on the probabilities $(p(d_1|e^t_{i_1}),\ldots,p(d_M|e^t_{i_1}))^T$, and $\mathscr{I}_{true} = 1$, $\mathscr{I}_{false} = 0$. The above must-link cost function means that if the label of $e^t_{i_1}$ is not equal to the label of $e^t_{i_2}$, then we should take into account the cost function of how dissimilar the two entities $e^t_{i_1}$ and $e^t_{i_2}$ are. The dissimilarity is computed based on the probability of document $\mathscr{D}$ given the entities $e^t_{i_1}$ and $e^t_{i_2}$ as Eq. (4). The more dissimilar the two entities are, the larger cost is imposed.

For cannot-links, the cost function is defined as

$$V_{\mathscr{C}}(e^t_{i_1}, e^t_{i_2} \in \mathscr{C}_{e^t_{i_1}}) = w_{\mathscr{C}}(D^t_{\max} - D_{KL}(p(\mathscr{D}|e^t_{i_1})\|p(\mathscr{D}|e^t_{i_2}))) \cdot \mathscr{I}_{l_{e^t_{i_1}} \neq l_{e^t_{i_2}}}, \quad (5)$$

where $w_{\mathscr{C}}$ is the weight for cannot-links, and $D^t_{\max}$ is the maximum value for all the $D_{KL}(p(\mathscr{D}|e^t_{i_1})\|p(\mathscr{D}|e^t_{i_2}))$. The cannot-link cost function means that if the label of $e^t_{i_1}$ is equal to the label of $e^t_{i_2}$, then we should take into account the cost function of how similar they are.

Integrating the constraints for $\mathscr{L}_e1,\ldots,\mathscr{L}_eT$ to Eq. (3), the objective function of constrained HIN clustering is:

$$\begin{aligned}\mathscr{J}_{\text{CHINC}} = {} & D_{KL}(p(\mathscr{D},\mathscr{W})\|q(\mathscr{D},\mathscr{W})) \\ & + \sum_{t=1}^{T} D_{KL}(p(\mathscr{D},\mathscr{E}^t)\|q(\mathscr{D},\mathscr{E}^t)) \\ & + \sum_{t=1}^{T}\sum_{s=1}^{T} D_{KL}(p(\mathscr{E}^t,\mathscr{E}^s)\|q(\mathscr{E}^t,\mathscr{E}^s)) \\ & + \sum_{t=1}^{T}\sum_{e^t_{i_1}=1}^{V_t}\sum_{e^t_{i_2}\in\mathscr{M}_{e^t_{i_1}}} V_{\mathscr{M}}(e^t_{i_1}, e^t_{i_2} \in \mathscr{M}_{e^t_{i_1}}) \\ & + \sum_{t=1}^{T}\sum_{e^t_{i_1}=1}^{V_t}\sum_{e^t_{i_2}\in\mathscr{C}_{e^t_{i_1}}} V_{\mathscr{C}}(e^t_{i_1}, e^t_{i_2} \in \mathscr{C}_{e^t_{i_1}}).\end{aligned} \quad (6)$$

From this objective function we can see that, the must-links and cannot-links are imposed to the entities that the semantic parsing detects. Since the task is document clustering, the subtypes of entities serve as indirect supervision because they cannot directly affect the cluster labels of the documents. However, the constraints can affect the labels of entities, and then the labels of entities can be transferred to the document side to affect the labels of documents.

### 3.2 Alternating Optimization

Since global optimization of all the latent labels as well as the approximate function $q(\cdot, \cdot)$ is intractable, we perform an alternating optimization shown in Algorithm 1. We iterate the process to optimize the labels of documents, words, and entities. Meanwhile, we update the function $q(\cdot, \cdot)$ for the corresponding types.

For example, to find label $l_{d_m}$ of document $d_m$, we have:

$$l_{d_m} = \arg\min_{l_{d_m}=k_d} D_{KL}(p(\mathscr{W}|d_m)\|p(\mathscr{W}|\hat{d}_{k_d})) + \sum_{t=1}^{T} D_{KL}(p(\mathscr{E}^t|d_m)\|p(\mathscr{E}^t|\hat{d}_{k_d})). \quad (7)$$

To find label $l_{w_i}$ of word $w_i$, we have:

$$l_{w_i} = \arg\min_{l_{w_i}=k_w} D_{KL}(p(\mathscr{D}|w_i)\|p(\mathscr{D}|\hat{w}_{k_w})). \quad (8)$$

**Algorithm 1**: Alternating Optimization for CHINC.

```
**Input:** HIN defined on documents $\mathscr{D}$, words $\mathscr{W}$, and entities $\mathscr{E}^t$, $t = 1, \dots, T$; Set maxIter and max$\delta$.
**while** iter < maxIter and $\delta >$ max$\delta$ **do**
    $\mathscr{D}$ **Label Update**: minimize Eq. (7) w.r.t. $\mathscr{L}_d$.
    $\mathscr{D}$ **Model Update**: update $q(d_m, w_i)$ and $q(d_m, e_i^t)$.
**for** $t = 1, \dots, T$ **do**
        $\mathscr{E}^t$ **Label Update**: minimize Eq. (9) w.r.t. $\mathscr{L}_{e^t}$.
        $\mathscr{E}^t$ **Model Update**: update $q(d_m, e_i^t)$ and $q(e_j^s, e_i^t)$.
    **end for**
    $\mathscr{D}$ **Label Update**: minimize Eq. (7) w.r.t. $\mathscr{L}_d$.
    $\mathscr{D}$ **Model Update**: update $q(d_m, w_i)$ and $q(d_m, e_i^t)$.
    $\mathscr{W}$ **Label Update**: minimize Eq. (8) w.r.t. $\mathscr{L}_w$.
    $\mathscr{W}$ **Model Update**: update $q(d_m, w_i)$.
    Compute cost change $\delta$ using Eq. (6).
**end while**
```

To find the label $l_{e_i^t}$, we use the iterated conditional mode (ICM) algorithm [3] to iteratively assign a label to the entity. We update one label $l_{e_i^t}$ at a time, and keep all the other labels fixed:

$$\begin{aligned} l_{e_i^t} = \arg\min_{l_{e_i^t}=k_{e^t}} & D_{KL}(p(\mathscr{D}|e_i^t)\|p(\mathscr{D}|\widehat{e^t}_{k_{e^t}})) \\ & + \sum_{s=1}^{T} D_{KL}(p(\mathscr{E}^s|e_i^t)\|p(\mathscr{E}^s|\widehat{e^t}_{k_{e^t}})) \\ & + \sum_{\substack{e_{i'}^t \in \mathscr{M}_{e_i^t}; \\ I_{l_{e_i^t} \neq l_{e_{i'}^t}}}} w_{\mathscr{M}} D_{KL}(p(\mathscr{D}|e_i^t)\|p(\mathscr{D}|e_{i'}^t)) \\ & + \sum_{\substack{e_{i'}^t \in \mathscr{C}_{e_i^t}; \\ I_{l_{e_i^t} = l_{e_{i'}^t}}}} w_{\mathscr{C}} (D_{\max}^t \\ & - D_{KL}(p(\mathscr{D}|e_i^t)\|p(\mathscr{D}|e_{i'}^t))). \end{aligned} \quad (9)$$

To transfer the original objective function (6) to Eq. (9), we should follow Eq. (2) where we replace the document and word notations to the entity notations. To understand why Eq. (2) holds, we suggest to refer to the original ITCC for detailed derivation [7].

Then, with the labels $\mathscr{L}_d$, $\mathscr{L}_{e_t}$ and $\mathscr{L}_w$ fixed, we update the model function $q(d_m, w_i)$, $q(d_m, e_i^t)$, and $q(e_j^s, e_i^t)$. The update of $q$ is not influenced by the must-links and cannot-links. Thus we can modify them the same as ITCC [7] and only show the update of $q(d_m, e_i^t)$ here:

$$q(\hat{d}_{k_d}, \widehat{e^t}_{k_{e^t}}) = \sum_{l_{d_m}=k_d} \sum_{l_{e_i^t}=k_{e^t}} p(d_m, e_i^t); \quad (10)$$

$$q(d_m|\hat{d}_{k_d})=\frac{q(d_m)}{q(l_{d_m}=k_d)}[q(d_m|\hat{d}_{k_d})=0 \text{ if } l_{d_m} \neq k_d]; \quad (11)$$

$$q(e_i^t|\widehat{e^t}_{k_{e^t}})=\frac{q(e_i^t)}{q(l_{e_i^t}=k_{e^t})}[q(e_i^t|\widehat{e^t}_{k_{e^t}})=0 \text{ if } l_{e_i^t} \neq k_{e^t}]; \quad (12)$$

where $q(d_m)=\sum_{e_i^t}p(d_m, e_i^t), q(e_i^t)=\sum_{d_m}p(d_m, e_i^t), q(\hat{d}_{k_d})=\sum_{k_{e^t}}p(\hat{d}_{k_d}, \widehat{e^t}k_{e^t})$ and $q(\widehat{e^t}_{k_{e^t}})=\sum_{k_d}p(\hat{d}_{k_d}, \widehat{e^t}_{k_{e^t}})$.

Algorithm 1 summarizes the main steps in the procedure. The objective function (6) with our alternating update monotonically decreases to a local optimum. This is because the ICM algorithm decreases the non-negative objective function (6) to a local optimum given a fixed *q* function. Then the update of *q* is monotonically decreasing as guaranteed by the theorem proven in [31].

The time complexity of Algorithm 1 is $O(n_{\mathscr{D},\mathscr{W}}\cdot(K_d+K_w)+\sum_{t=1}^{T}n_{\mathscr{D},\mathscr{E}^t}\cdot(K_d+K_{e^t})+\sum_{t=1}^{T}\sum_{s=1}^{T}(n_{\mathscr{E}^t,\mathscr{E}^s}+(n_c * iter_{ICM}))\cdot(K_{e^t}+K_{e^s}))\cdot iter_{AO}$, where $n_{\cdot,\cdot}$ is the total number of nonzero elements in the corresponding co-occurrence matrix, $n_c$ is the number of constraints, $iter_{ICM}$ is the number of ICM iterations, $K_d$, $K_w$ and $K_e t$ are the number of document clusters, word clusters and entity clusters of type *t*, and $iter_{AO}$ is the number of the alternating optimization iterations.

## 4. Experiments

In this section, we show the experimental results to demonstrate the effectiveness and efficiency of our approach on document clustering with world knowledge as indirect supervision.

### 4.1 Datasets

We use the following two benchmark datasets to evaluate domain dependent document clustering. For both datasets we assume the numbers of document clusters are given.

**20Newsgroups (20NG):** The 20newsgroups dataset contains about 20,000 newsgroups documents evenly distributed across 20 newsgroups.[2] We use all the 20 groups as 20 classes.

**RCV1:** The RCV1 dataset is a dataset containing manually labeled newswire stories from Reuter Ltd [19]. The news documents are categorized with respect to three controlled vocabularies: industries, topics and regions. There are 103 categories including all nodes except for root in the hierarchy. The maximum depth is four, and 82 nodes are leaves. We select top categories MCAT (Markets), CCAT (Corporate/Industrial) and ECAT

[2] http://qwone.com/~jason/20Newsgroups/

(Economics) in one portion of the test partition to form three clustering tasks. The three clustering tasks are summarized in Table 2. We use the original source of this data, and use the leaf categories in each task as the ground-truth classes.

### 4.2 World Knowledge Bases

Then we introduce the knowledge bases we use.

**Freebase:** Freebase[3] is a publicly available knowledge base consisting of entities and relations collaboratively collected by its community members. Now, it contains over 2 billions relation expressions between 40 millions entities. We convert a logical form generated by our unsupervised semantic parser of the world knowledge specification approach introduced in Section 2.1 into a SPARQL query and execute it on our copy of Freebase using the Virtuoso engine.

**YAGO2:** YAGO2[4] is also a semantic knowledge base, derived from Wikipedia, WordNet and GeoNames. Currently, YAGO2 has knowledge of more than 10 million entities (like persons, organizations, cities, etc.) and contains more than 120 million facts about these entities. Similar to Freebase, we also convert a logical form into a SPARQL query and execute it on our copy of YAGO2 using the Virtuoso engine.

In Table 3, we show some statistics about Freebase and YAGO2.

Note that in most knowledge bases, such as Freebase and YAGO2, entities types are often organized in a hierarchical manner. For example, *Politician* is a sub-type of *Person. University* is a subtype of *Organization*. All the types or attributes share a common root, called *Object*. Figure 3 depicts an example of hierarchy of types. In general, we use the highest level under the root object as the entity types (e.g., *Person*) as specified world knowledge incorporated in the HIN, and the direct children (e.g., *Politician*) as entity constraints. In the following experiments, we select *Person*, *Organization*, and *Location* as the three entity types in the HIN, because they are popular in both Freebase and YAGO2.

### 4.3 Effectiveness of World Knowledge Specification

Before applying the specified world knowledge to downstream text analytics tasks, such as document clustering in our case, we need to evaluate whether our world knowledge specification approach could produce the correct specified world knowledge.

In order to test the effectiveness of our world knowledge specification approach, we first sample 200 documents from 20 newsgroups, i.e., 10 documents from each category. Second, we split the documents into sentences. After post-processing, 3,232 sentences are generated for human evaluation. Third, we use our world knowledge specification approach in Section 2.1 with three different semantic filtering modules to generate the specified world knowledge for each sentence, which consists of relation triplets in the form of ($e_1$, $r$, $e_2$) with the type information. Afterwards, we ask three annotators to label the specified world

[3] https://developers.google.com/freebase/

[4] http://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/

knowledge according two criterion: (1) whether the boundaries of $e_1$ and $e_2$ are correctly recognized or not; (2) whether the entity type of $e_1$ and $e_2$ are correct or not. It is annotated as correct if both (1) and (2) are satisfied. We check the mutual agreement of the human annotation, which is around 91.3% accuracy.

We then test the precision of three different specified world knowledge generated by the corresponding semantic filtering method. The results are shown in Table 4. From the results we can see that, CBSF outpeforms the other two ways to generate the correct semantic meaning. The main reason is that, conceptualization based method is able to use the context information to help judge the real semantic of the text rather than only taking the statistics of the data into account. Here we only care about precision because we wish to use world knowledge as indirect supervision. The recall will not be very important.

### 4.4 Clustering Result

In this experiment, we compare the performance of our model, constrained heterogeneous information network clustering (CHINC), with several representative clustering algorithms such as Kmeans, ITCC [7] and CITCC [31]. The parameters used in CHINC to control the constraints are $w_{\mathscr{M}}$ and $w_{\mathscr{C}}$. We set them following the rules tested in [31]. We also denote our algorithm without constraints as HINC. “FB” and “YG” represent two different world knowledge sources, Freebase and YAGO2, respectively. We re-implement all the above clustering algorithms. Notice that, for CITCC, we follow [31] to generate and add constraints for documents and words. We also use the specified world knowledge as features to enhance the Kmeans and ITCC. The feature settings are defined as below:

- BOW: Traditional bag-of-words model with the tf-idf weighting mechanism.
- BOW+FB: BOW integrated with additional features from entities in specified world knowledge of Freebase.
- BOW+YG: BOW integrated with additional features from entities in specified world knowledge of YAGO2.

We employ the widely-used normalized mutual information (NMI) [34] as the evaluation measure. The NMI score is 1 if the clustering results match the category labels perfectly and 0 if the clusters are obtained from a random partition. In general, the larger the scores are, the better the clustering results are.

In Table 5, we show the performance of all the clustering algorithms with different experimental settings. The NMI is the average NMI of five random trials per experiment setting. Overall, among all the methods we test, CHINC consistently performs the best among all the clustering methods. We can see that HINC+FB and HINC+YG perform better than ITCC with BOW+FB or BOW+YG features, respectively. This means that by using the structural information provided by the world knowledge, we can further improve the clustering results. In addition, the algorithms with Freebase consistently outperform the ones with YAGO2, since Freebase has much more facts compared with YAGO2 as shown in Table 3; besides, one can see in Figure 4 that Freebase could consistently specify more entities than YAGO2 does from all of the document datasets. CITCC is the strongest baseline clustering algorithm, because it uses the ground-truth constraints derived from category labels based on the human knowledge. We use 250K constraints to perform CITCC. As shown in Table 5, HINC performs competitive with the CITCC. CHINC significantly outperforms CITCC. This shows that by automatically using world knowledge, it has the potential to perform better than the algorithm with the specific domain knowledge.

#### 4.4.1 Analysis of Number of Entity Clusters—

We also evaluate the effect of varying the number of entity clusters of each entity type in CHINC on the document clustering task. Figure 5 shows the results of clustering with different numbers of entity clusters of each entity type on “CHINC + Freebase” for the 20NG dataset. The number of entity clusters varies from 2 to 128. The default number of iterations is set as 20, which will be discussed in Section 4.4.2. When testing the effect of the number of entity clusters of one entity type, the numbers of entity clusters of the other two entity types are fixed as twice as the number of document clusters, which are 40 and 40 in 20NG, respectively. It is shown that for this dataset, more entity clusters may not result in improved document clustering results when a sufficient number of entity clusters is reached. For example, as shown in Figure 5, after reaching 32, the NMI scores of CHINC actually decrease when the numbers of entity clusters further increase. From the results, we can conclude that, there exist certain values of the number of entity clusters leading to the best clustering peformance. Similar to the results on “CHINC + Freebase” for 20NG dataset, in the rest of the experiments, we fix the number of entity clusters of each entity type to be twice the number of document clusters.

#### 4.4.2 Analysis of Number of Iterations in Alternating Optimization—

We evaluate the impact of the number of iterations of the alternating optimization (Algorithm 1) on CHINC in relation to the execution time of the optimization algorithm as well as the clustering performance. We increase the number of iterations from 1 to 80. For example, for each number of iterations, we run CHINC five trials, and the average execution time and NMI are summarized in Figure 6. From the result, one can conclude that the larger number of iterations is, the more significant the improvement on clustering performance. This improvement eventually drops, tapers out, and becomes stable. The reason is that, along with the increase of the number of iterations, the alternating optimization algorithm comes to covergence. However, the execution time still increase in a nearly linear manner. For example, as shown in Figure 6, after reaching 20, the performance stays stable. Thus, we set the number of iterations as 20 in the remaining experiments with the consideration of both performance and efficiency. Similarly, we set the number of iterations as 20 when conducting experiments on the other combinations of document datasets and world knowledge bases.

#### 4.4.3 Analysis of Specified World Knowledge based Constraints—

Rather than using human knowledge as constraints, we use the specified world knowledge automatically generated by our approach as constraints in CHINC. Based on the specified world knowledge, it is straightforward to design constraints for entities.

***Entity constraints:*** **(1) Must-links.** If two entities belong to the same entity sub-type, we add a must-link. **(2) Cannot-links.** If two entities belong to different entity sub-types, we add a cannot-link. For example, the entity sub-types of “Obama” and “United States” are *Politician* and *Country* respectively. In this case, we add a cannot-link to them.

We then test the performance of our proposed CHINC by using the specified world knowledge as constraints described above. We show the experiments on “CHINC + Freebase” for 20NG dataset in Figure 7. Each *x*-axis represents the number of entity type constraints used in each experiment, and *y*-axis is the average NMI of five random trials. The constraints derived from entity type #1, #2, and #3 are eventually added to CHINC as shown in Figure 7a, Figure 7b and Figure 7c, respectively. We can see that CHINC outperforms the best clustering algorithm with the human knowledge as shown in Table 5 (CITCC: 0.569) with even no constraints (HINC: 0.571). By adding more and more constraints, the clustering result of CHINC is significantly better. So CHINC is able to use information in world knowledge specified in the HIN, and the entity sub-type information can be transferred to the document side. The results show the power of modeling data as heterogeneous information networks, as well as the high quality of constraints derived from world knowledge.

From Figure 7, by increasing the number of constraints, we find that the average execution time of five trials increases linearly, and the clustering performance measured by NMI is increasing as mentioned before. Figure 7c shows the effects of the constraints of all the three entity types on the clustering performance as well as the execution time. After the number of constraints reach 50M, the increase of performance drops and stays stable. At this point, the execution time is around 1.2M (ms). In Figure 8, we can see the similar results on the other combinations of document datasets and knowledge bases. We also find that the average execution time of our algorithm with Freebase as world knowledge source is greater than that with YAGO2. As shown in Figure 4, the reason is that each document datasets with Freebase could be specified much more entities than that with YAGO2. From the results, we can see that our algorithm is scalable to use the large scale specified world knowledge as constraints, and cluster large amounts of documents.

## 5. Related Work

In this section, we review the related work on machine learning with world knowledge and heterogeneous information networks, and have a detailed discussion on them.

### 5.1 Machine Learning with World Knowledge

Most of the existing usage of world knowledge is to enrich the features beyond bag-of-words representation of documents. For example, by using the linguistic knowledge base WordNet to resolve synonyms and introduce WordNet concepts, the quality of document clustering can be improved [14]. The first paper using the term “world knowledge” [11] extends the bag-of-words features with the categories in Open Directory Project (ODP), and shows that it can help improve text classification with additional knowledge. Following this, by mapping the text to the semantic space provided by Wikipedia pages, it has been proven to be useful for short text classification [12] and clustering [15, 16]. Liu et al. [22] also use another knowledge base of taxonomy, Probase, to enrich the features of ads keywords to build a new taxonomy of domain dependent keyword set. All of the above approaches just consider to use world knowledge as a source of features. However, the knowledge in the knowledge bases indeed has annotations of types, categories, etc.. Thus, it can be more effective to consider this information as “supervision” to supervise other machine learning algorithms and tasks.

Distant supervision uses the knowledge of entities and their relationships from world knowledge bases, e.g., Freebase, as supervision for the task of entity and relation extraction [25, 39, 41]. It considers to use knowledge supervision to extract more entities and relations from new text or to generate a better embedding of entities and relations. Thus, the application of direct supervision is limited to entities and relations themselves.

Song et al. [31] consider using fully unsupervised method to generate constraints of words using an external general-purpose knowledge base, WordNet. This can be regarded as an initial attempt to use general knowledge as indirect supervision to help clustering. However, the knowledge from WordNet is mostly linguistically related. It lacks of the information about named entities and their types. Moreover, their approach is still a simple application of constrained co-clustering, where it misses the rich structural information in the knowledge base.

### 5.2 Heterogeneous Information Network

A heterogeneous information network (HIN) is defined as a graph of multi-typed entities and relations [13]. Different from traditional graphs, HIN incorporates the type information which can be useful to identify the semantic meaning of the paths in the graph [36]. This is a good property to perform graph search and matching. Original HINs are developed for the applications of scientific publication network analysis [36, 37]. Then social network analysis also leverages this representation for user similarity and link prediction [42]. Seamlessly, we can see that the knowledge in world knowledge bases, e.g., Freebase and YAGO2, can be naturally represented as an HIN, since the entities and relations in the knowledge base are all typed. We introduce this representation to knowledge based analysis, and show that it can be very useful for our document clustering task. Note that there is also a series of methods called multi-type relational data clustering [23, 24]. While they require the data to be structural beforehand (e.g., providing information of authors, co-authors, etc.), our method only needs the input of raw documents. In addition to the multi-type relational information, we also incorporate the type information provided by the knowledge base as constraints to further improve the clustering results.

## 6. Conclusion

In this paper, we study a novel problem of machine learning with world knowledge. Particularly, we take document clustering as an example and show how to use world knowledge as indirect supervision to improve the clustering results. To use the world knowledge, we show how to adapt the world knowledge to domain dependent tasks by using semantic parsing and semantic filtering. Then we represent the data as a heterogeneous information network, and use a constrained network clustering algorithm to obtain the document clusters. We demonstrate the effectiveness and efficiency of our approach on two real datasets along with two popular knowledge bases. In the future, we plan to use world knowledge to help more text mining and text analytics tasks, such as text classification and information retrieval.

## Acknowledgments

Chenguang Wang gratefully acknowledges the support by the National Natural Science Foundation of China (NSFC Grant Number 61472006) and the National Basic Research Program (973 Program No. 2014CB340405). The research is also partially supported by the Army Research Laboratory (ARL) under agreement W911NF-09-2-0053, and by DARPA under agreement number FA8750-13-2-0008. Research is also partially sponsored by 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), and MIAS, a DHS-IDS Center for Multimodal Information Access and Synthesis at UIUC. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied by these agencies or the U.S. Government.

## References

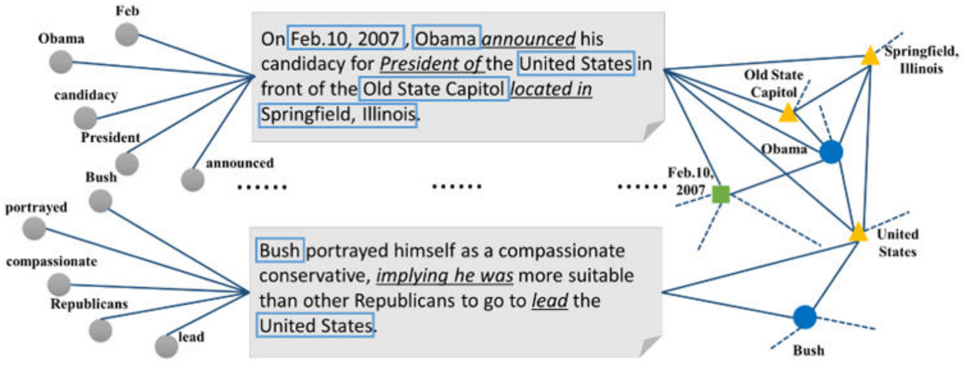
1. Auer, S.; Bizer, C.; Kobilarov, G.; Lehmann, J.; Cyganiak, R.; Ives, Z. Dbpedia: A nucleus for a web of open data. Springer; 2007.
2. Banko M, Cafarella MJ, Soderland S, Broadhead M, Etzioni O. Open information extraction from the web. IJCAI. 2007:2670–2676.
3. Basu S, Bilenko M, Mooney RJ. A probabilistic framework for semi-supervised clustering. KDD. 2004:59–68.
4. Berant J, Chou A, Frostig R, Liang P. Semantic parsing on freebase from question-answer pairs. EMNLP. 2013:1533–1544.
5. Bollacker KD, Evans C, Paritosh P, Sturge T, Taylor J. Freebase: a collaboratively created graph database for structuring human knowledge. SIGMOD. 2008:1247–1250.
6. Chapelle, O.; Schölkopf, B.; Zien, A., editors. Semi-Supervised Learning. MIT Press; 2006.
7. Dhillon IS, Mallela S, Modha DS. Information-theoretic co-clustering. KDD. 2003:89–98.
8. Dong X, Gabrilovich E, Heitz G, Horn W, Lao N, Murphy K, Strohmann T, Sun S, Zhang W. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. KDD. 2014:601–610.
9. Etzioni O, Cafarella M, Downey D. Webscale information extraction in knowitall (preliminary results). WWW. 2004:100–110.
10. Fellbaum, C., editor. WordNet: an electronic lexical database. MIT Press; 1998.
11. Gabrilovich E, Markovitch S. Feature generation for text categorization using world knowledge. IJCAI. 2005:1048–1053.
12. Gabrilovich E, Markovitch S. Computing semantic relatedness using wikipedia-based explicit semantic analysis. IJCAI. 2007:1606–1611.
13. Han J, Sun Y, Yan X, Yu PS. Mining knowledge from databases: An information network analysis approach. SIGMOD. 2010:1251–1252.
14. Hotho A, Staab S, Stumme G. Ontologies improve text document clustering. ICDM. 2003:541–544.
15. Hu J, Fang L, Cao Y, Zeng HJ, Li H, Yang Q, Chen Z. Enhancing text clustering by leveraging Wikipedia semantics. SIGIR. 2008:179–186.
16. Hu X, Zhang X, Lu C, Park EK, Zhou X. Exploiting wikipedia as external knowledge for document clustering. KDD. 2009:389–396.
17. Kwiatkowski T, Zettlemoyer LS, Goldwater S, Steedman M. Lexical generalization in CCG grammar induction for semantic parsing. EMNLP. 2011:1512–1523.
18. Lenat, DB.; Guha, RV. Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project. Addison-Wesley; 1989.
19. Lewis DD, Yang Y, Rose TG, Li F. Rcv1: A new benchmark collection for text categorization research. JMLR. 2004; 5:361–397.

20. Li Y, Wang C, Han F, Han J, Roth D, Yan X. Mining evidences for named entity disambiguation. KDD. 2013:1070–1078.
21. Liang P. Lambda dependency-based compositional semantics. arXiv. 2013
22. Liu X, Song Y, Liu S, Wang H. Automatic taxonomy construction from keywords. KDD. 2012:1433–1441.
23. Long B, Zhang ZM, Wú X, Yu PS. Spectral clustering for multi-type relational data. ICML. 2006:585–592.
24. Long B, Zhang ZM, Yu PS. A probabilistic framework for relational clustering. KDD. 2007:470–479.
25. Mintz M, Bills S, Snow R, Jurafsky D. Distant supervision for relation extraction without labeled data. ACL/AFNLP. 2009:1003–1011.
26. Mitchell TM, Cohen WW, H ER Jr, Talukdar PP, Betteridge J, Carlson A, Mishra BD, Gardner M, Kisiel B, Krishnamurthy J, Lao N, Mazaitis K, Mohamed T, Nakashole N, Platanios EA, Ritter A, Samadi M, Settles B, Wang RC, Wijaya DT, Gupta A, Chen X, Saparov A, Greaves M, Welling J. Never-ending learning. AAAI. 2015:2302–2310.
27. Mooney RJ. Learning for semantic parsing. CICLing. 2007:311–324.
28. Pan SJ, Yang Q. A survey on transfer learning. IEEE TKDE. 2010; 22(10):1345–1359.
29. Ponzetto SP, Strube M. Deriving a large-scale taxonomy from wikipedia. AAAI. 2007:1440–1445.
30. Ratinov L, Roth D. Design challenges and misconceptions in named entity recognition. CoNLL. 2009:147–155.
31. Song Y, Pan S, Liu S, Wei F, Zhou M, Qian W. Constrained text coclustering with supervised and unsupervised constraints. IEEE TKDE. 2013; 25(6):1227–1239.
32. Song Y, Wang H, Wang Z, Li H, Chen W. Short text conceptualization using a probabilistic knowledgebase. IJCAI. 2011:2330–2336.
33. Song Y, Wang S, Wang H. Open domain short text conceptualization: A generative + descriptive modeling approach. IJCAI. 2015
34. Strehl A, Ghosh J. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. JMLR. 2003; 3:583–617.
35. Suchanek FM, Kasneci G, Weikum G. Yago: a core of semantic knowledge. WWW. 2007:697–706.
36. Sun Y, Han J, Yan X, Yu PS, Wu T. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. PVLDB. 2011:992–1003.
37. Sun Y, Norick B, Han J, Yan X, Yu PS, Yu X. Integrating meta-path selection with user-guided object clustering in heterogeneous information networks. KDD. 2012:1348–1356.
38. Wang C, Song Y, Roth D, Wang C, Han J, Ji H, Zhang M. Constrained information-theoretic tripartite graph clustering to identify semantically similar relations. IJCAI. 2015
39. Wang Z, Zhang J, Feng J, Chen Z. Knowledge graph and text jointly embedding. EMNLP. 2014:1591–1601.
40. Wu W, Li H, Wang H, Zhu KQ. Probase: A probabilistic taxonomy for text understanding. SIGMOD. 2012:481–492.
41. Xu C, Bai Y, Bian J, Gao B, Wang G, Liu X, Liu TY. Rc-net: A general framework for incorporating knowledge into word representations. CIKM. 2014:1219–1228.
42. Zhang J, Kong X, Yu PS. Transferring heterogeneous links across location-based social networks. WSDM. 2014:303–312.

## General Terms

### Algorithms

### Experimentation



On Feb.10, 2007, Obama *announced* his candidacy for *President of* the United States in front of the Old State Capitol *located in* Springfield, Illinois.

Bush portrayed himself as a compassionate conservative, *implying he was* more suitable than other Republicans to go to *lead* the United States.

**Figure 1.**

Heterogeneous information network example. The network $\mathcal{G}$ contains five entity types: document, word, date, person and location, which are represented with gray rectangle, gray round, green square, blue round, and yellow triangle, respectively.

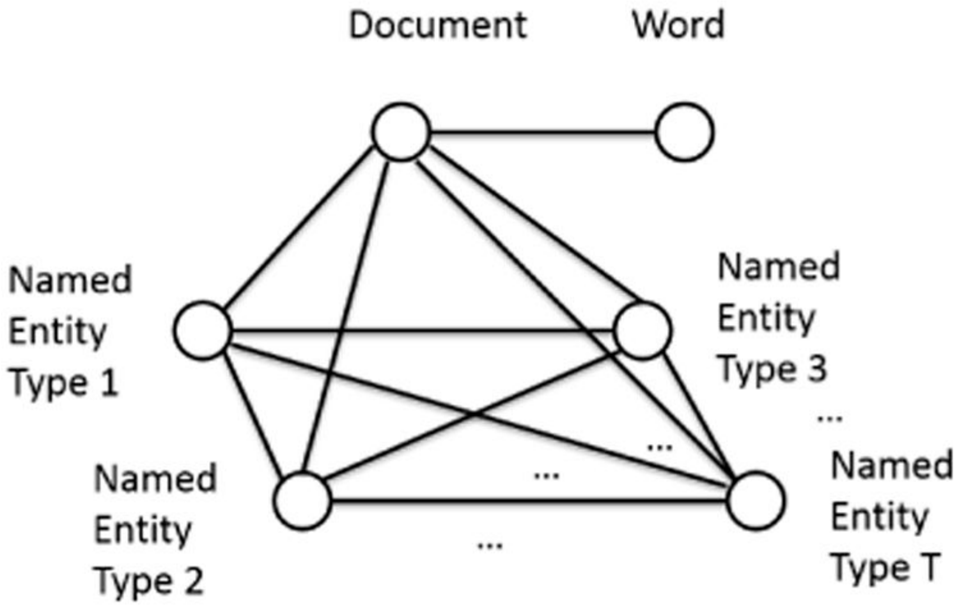


**Figure 2.**

Heterogeneous information network schema. The specified knowledge is represented in the form of heterogeneous information network. The schema contains multiple entity types: document $\mathscr{D}$, word $\mathscr{W}$, named entities $\{\mathscr{E}^{\mathrm{I}}\}_{I=1}^{T}$, and the relation types connecting the entity types.

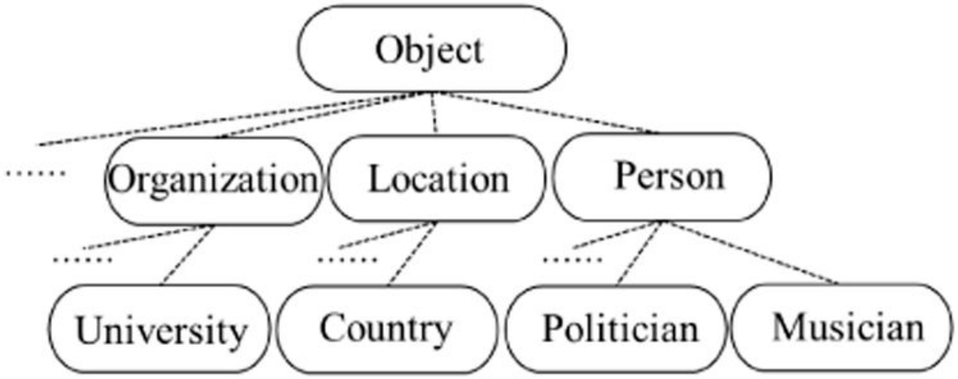


**Figure 3.**
Hierarchy of entity types.

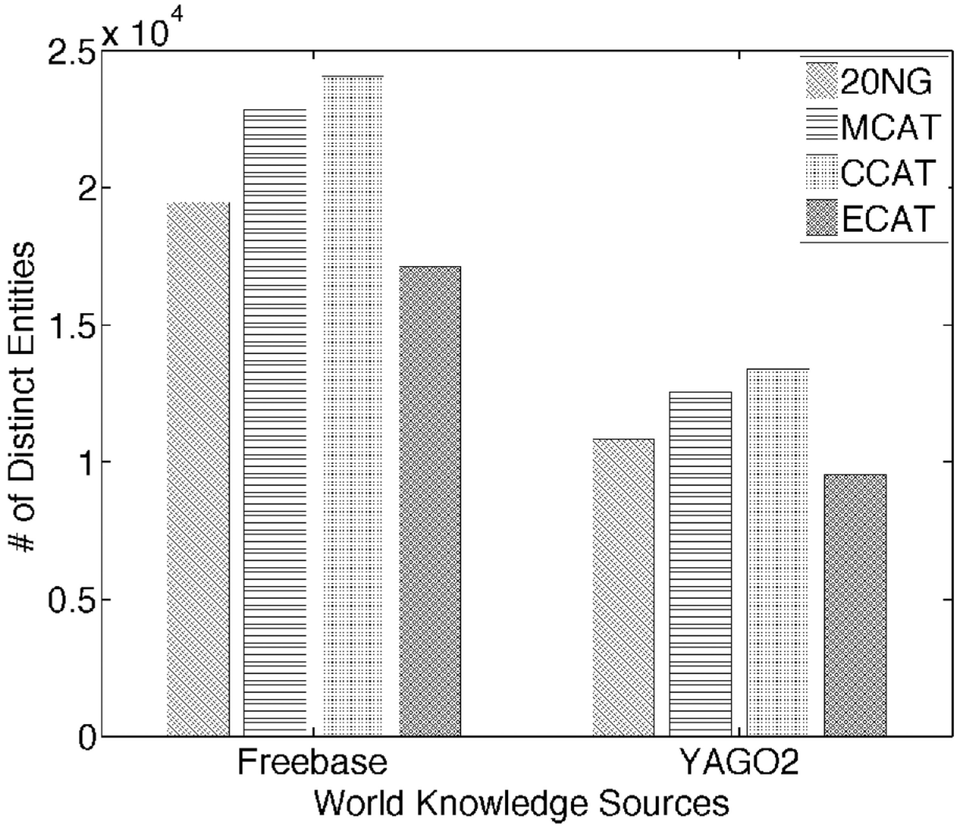


**Figure 4.**

Statistics of the number of entities in different document datasets with different world knowledge sources.

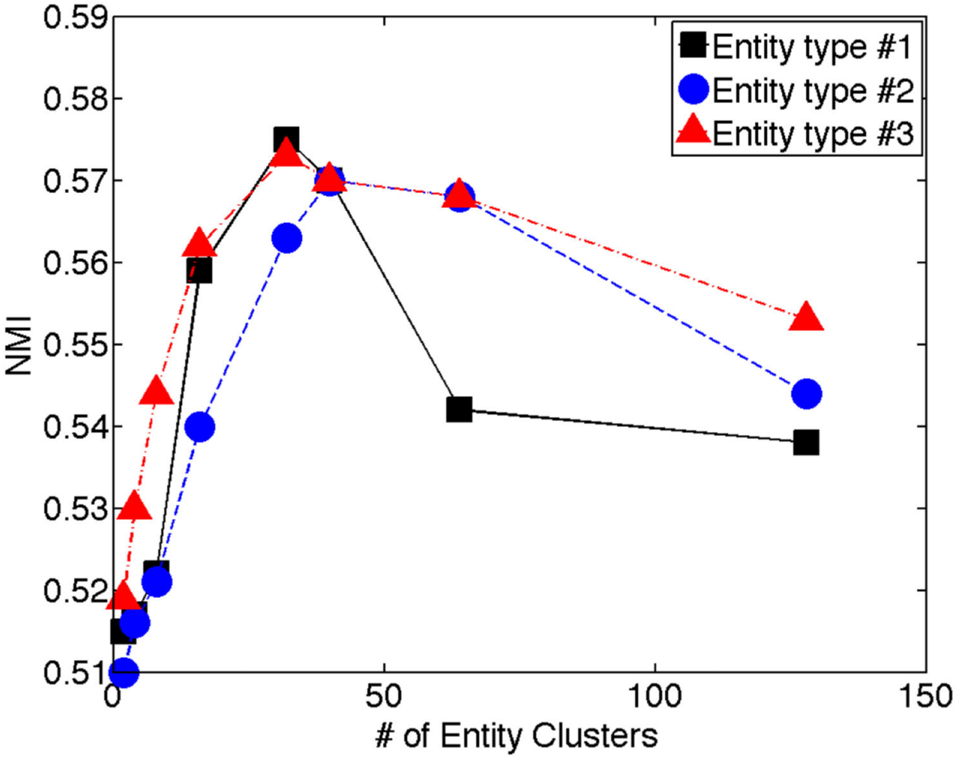


**Figure 5.**

Effect of the number of entity clusters of each entity type on document clustering on “CHINC + Freebase” for 20NG dataset.

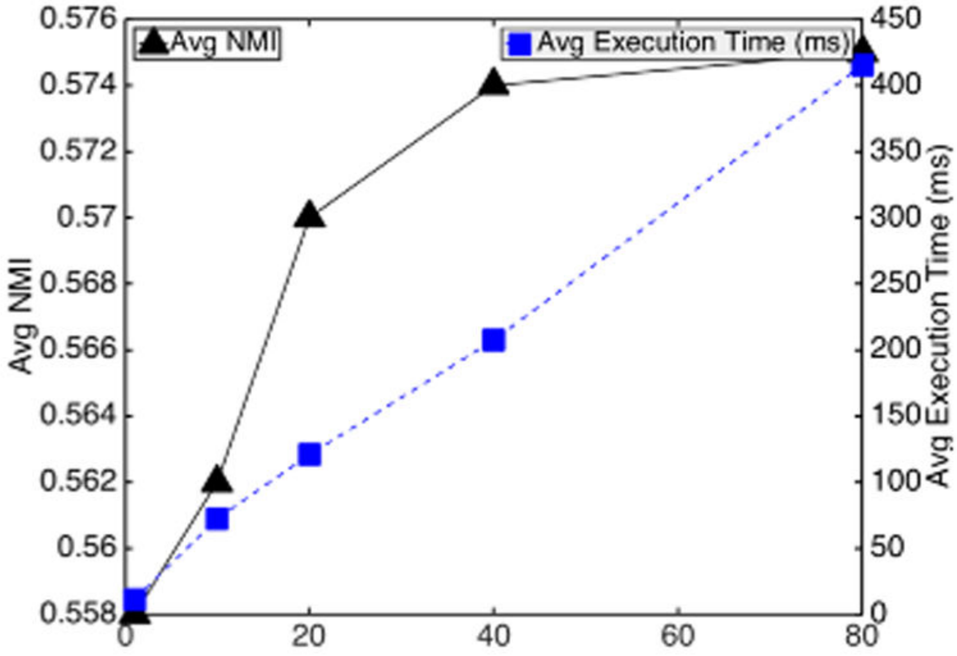
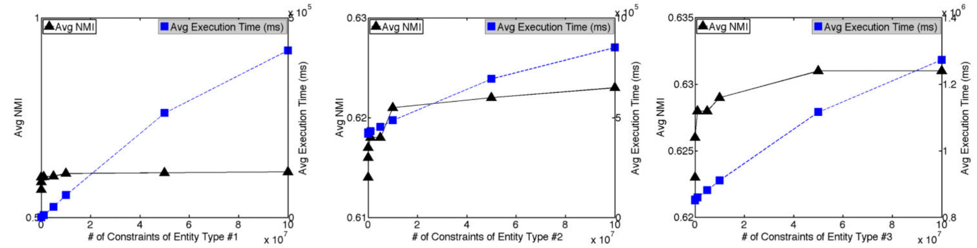


**Figure 6.**

Analysis of # of iterations in alternating optimization algorithm on “CHINC + Freebase” for 20NG dataset. Left *y*-axis: average NMI; Right *y*-axis: average execution time (ms).



(a) Effects of entity constraints of type #1.

(b) Effects of entity constraints of types #1+#2.

(c) Effects of entity constraints of types #1+#2+#3.

**Figure 7.**

Effects of entity constraints of "CHINC + Freebase" for 20NG dataset. Left *y*-axis: average NMI; Right *y*-axis: average execution time (ms).

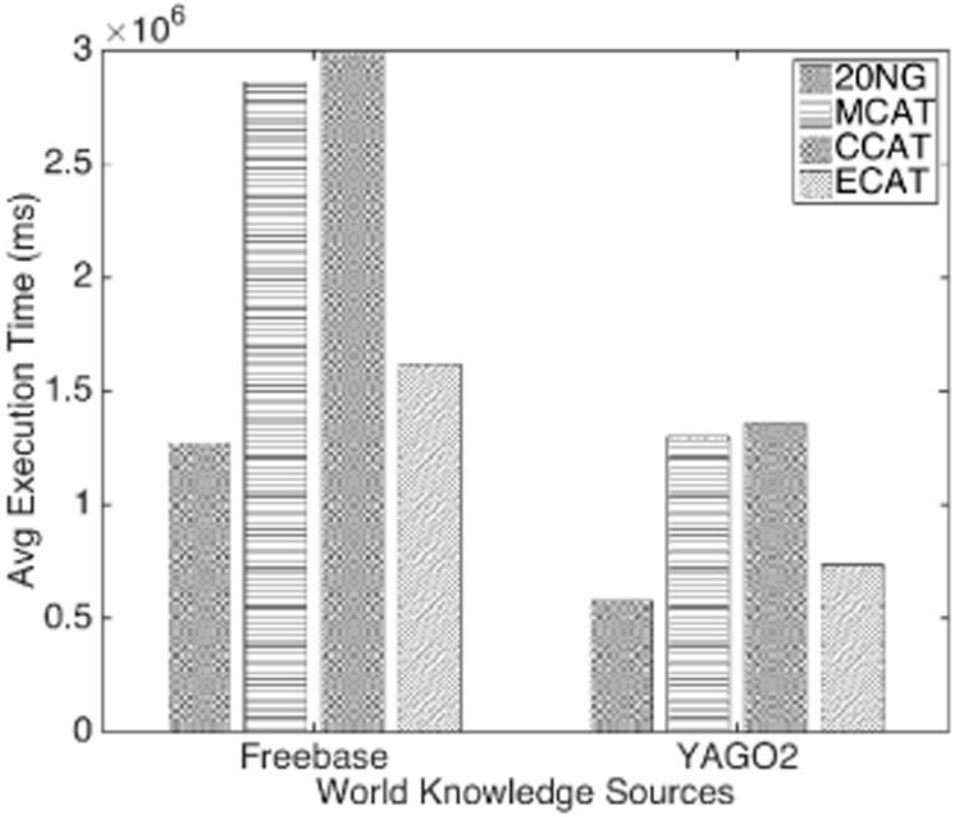


**Figure 8.**

Analysis of the efficiency of our algorithm on different document datasets with different world knowledge sources.

## Table 1

Notations for clustering algorithm. The indicators are used for the probability representation, while the indices are used as ids for the clusters.

| Meaning | Document | Word | Named Entity |
|---|---|---|---|
| Cluster Index | $k_d$ | $k_w$ | $k_{e^t}$ |
| Cluster Indicator | $d_{\widehat{k_d}}$ | $\hat{W}_{k_w}$ | $\widehat{e^t}_{k_{e^t}}$ |
| Data Indicator | $d_m$ | $w_i$ | $e_i^t$ |
| Data Indicator Set | $\mathscr{D}$ | $\mathscr{W}$ | $\mathscr{E}^t$ |
| Label | $l_{d_m}$ | $l_{w_i}$ | $l_{e_i^t}$ |
| Label Indicator Set | $\mathscr{L}_d$ | $\mathscr{L}_w$ | $\mathscr{L}_{e^t}$ |

### Table 2

RCV1 dataset statistics. #(Categories) is the number of all categories; # (Leaf Categories) is the number of leaf categories; # (Documents) is the number of documents.

| | #(Categories) | #(Leaf Categories) | #(Documents) |
|---|---|---|---|
| MCAT | 9 | 7 | 44,033 |
| CCAT | 31 | 26 | 47,494 |
| ECAT | 23 | 18 | 19,813 |

## Table 3

Statistics of Freebase and YAGO2. #(Entity Types) is the number of entity types; # (Entity Instances) is the number of entity instances; # (Relation Types) is the number of relation types; # (Relation Instances) is the number of relation instances.

| Name | Freebase | YAGO2 |
|---|---|---|
| #(Entity Types) | 1,500[a] | 350,000 |
| #(Entity Instances) | 40 millions | 10 millions |
| #(Relation Types) | 35,000 | 100 |
| #(Relation Instances) | 2 billions | 120 millions |

[a] The number of 1,500 types is reported in [8]. In our downloaded dump of Freebase, we found 79 domains, 2,232 types, and 6,635 properties.

## Table 4

Precision of different semantic filtering results. FBSF represents frequency based semantic filter; DFBSF represents document frequency based semantic filter; CBSF represents conceptualization based semantic filter.

| Semantic Filter | FBSF | DFBSF | CBSF |
|---|---|---|---|
| Precision | 0.751 | 0.890 | 0.916 |

### Table 5

Performance of different clustering algorithms on 20NG and RCV1 data. CHINC is our proposed method. BOW, FB (Freebase), or YG (YAGO2) represent bag of word features, the entities generated by our world knowledge specification approach based on Freebase or YAGO2, respectively. We compared all the numbers of HINC and CHINC with CITCC, which is the strongest baseline. The percentage in the brackets are the relative number compared to CITCC. CITCC uses 250K constraints generated based on ground-truth labels of documents.

| | **Kmeans** | | | **ITCC** | | | **CITCC** | **HINC** | | **CHINC** | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Features Data** | **BOW** | **BOW +FB** | **BOW +YG** | **BOW** | **BOW +FB** | **BOW +YG** | **BOW** | **FB** | **YG** | **FB** | **YG** |
| 20NG | 0.429 | 0.447 | 0.437 | 0.501 | 0.525 | 0.513 | 0.569 | 0.571 (+**0.4**%) | 0.541 (−**4.9**%) | 0.631 (+**10.9**%) | 0.600 (+**5.5**%) |
| MCAT | 0.549 | 0.575 | 0.559 | 0.604 | 0.630 | 0.619 | 0.652 | 0.645 (−**1.1**%) | 0.625 (−**4.1**%) | 0.698 (+**7.1**%) | 0.685 (+**5.1**%) |
| CCAT | 0.403 | 0.419 | 0.410 | 0.472 | 0.494 | 0.481 | 0.535 | 0.542 (+**1.3**%) | 0.515 (−**3.7**%) | 0.606 (+**13.3**%) | 0.574 (+**7.3**%) |
| ECAT | 0.417 | 0.436 | 0.424 | 0.493 | 0.516 | 0.505 | 0.562 | 0.561 (−**0.2**%) | 0.530 (−**5.7**%) | 0.624 (+**11.0**%) | 0.588 (+**4.6**%) |