Concept Grounding to Multiple Knowledge Bases via Indirect Supervision

Chen-Tse Tsai and **Dan Roth**
University of Illinois at Urbana-Champaign
201 N. Goodwin, Urbana, Illinois, 61801
{ctsai12, danr}@illinois.edu

Abstract

We consider the problem of disambiguating concept mentions appearing in documents and grounding them in multiple knowledge bases, where each knowledge base addresses some aspects of the domain. This problem poses a few additional challenges beyond those addressed in the popular Wikification problem. Key among them is that most knowledge bases do not contain the rich textual and structural information Wikipedia does; consequently, the main supervision signal used to train Wikification rankers does not exist anymore. In this work we develop an algorithmic approach that, by carefully examining the relations between various related knowledge bases, generates an indirect supervision signal it uses to train a ranking model that accurately chooses knowledge base entries for a given mention; moreover, it also induces prior knowledge that can be used to support a global coherent mapping of all the concepts in a given document to the knowledge bases.

Using the biomedical domain as our application, we show that our indirectly supervised ranking model outperforms other unsupervised baselines and that the quality of this indirect supervision scheme is very close to a supervised model. We also show that considering multiple knowledge bases together has an advantage over grounding concepts to each knowledge base individually.

1 Introduction

Grounding entities and concepts appearing in text documents to a knowledge base (KB) has become a popular method for contextually disambiguating them and can be used also for focused knowledge acquisition. It has been shown a valuable component for several natural language processing and information extraction tasks across different domains. In the news domain, the task is often called *Wikification* or *Entity Linking* and has been studied extensively recently (Bunescu and Pasca, 2006; Cucerzan, 2007; Mihalcea and Csomai, 2007; Ratinov et al., 2011; Cheng and Roth, 2013). Wikipedia is widely used as the target KB due to its broad coverage and detailed information of concepts. While Wikipedia is an excellent general purpose encyclopedic resource, when the text is domain specific, it may not be the single ideal resource; the text could be better "covered" by multiple ontological or encyclopedic resources.

This is clearly the case for scientific text which is often covered by multiple ontologies, each addressing some aspects of the domain. For example, in the biological domain there are multiple ontologies: Entrez Gene (Lu and Wilbur, 2010) focuses on genomes that have been completely sequenced; Gene Ontology (Ashburner et al., 2000) more broadly describes gene product characteristics; and ChEBI, is a dictionary of molecular entities focused on "small" chemical compounds. The ontologies provide complementary information, but they overlap and, in these cases, make use of different vocabulary and provide different relevant information.

In this paper, we consider the problem of grounding concepts appearing in documents to multiple KBs. We use the biomedical domain as our appli-

Transactions of the Association for Computational Linguistics, vol. 4, pp. 141–154, 2016. Action Editor: Rada Mihalcea; Lillian Lee.
Submission batch: 3/2015; Revision batch: 1/2015; 12/2015; 1/2016; Published 4/2016.
©2016 Association for Computational Linguistics. Distributed under a CC-BY 4.0 license.
Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/tacl_a_00089 by guest on 22 March 2026

cation domain, both due to its importance and to the fact that thousands of person-years have been spent on putting together a large number of relevant KBs. We discuss other potential applications in the end of the paper. The challenges in this problem are due both to ambiguity and variability in expressing concepts: a given mention in text can be used to express different concepts in the KBs, and a KB (ontology) concept may be expressed in text in multiple ways, such as synonyms or nicknames. In the case of using multiple KBs, an additional challenge is due to the overlap between KBs: a mention can refer to multiple concepts in different KBs and we want to ground the mention to all of them. Figure 1 shows an example of concept annotations from the CRAFT corpus (Bada et al., 2012). The mention *BRCA2* refers both to "breast cancer type 2 susceptibility protein (PR:000004804)" from the Protein Ontology and to "BRCA2 (EG:675)" from the Entrez Gene database, which has more than one hundred genes across different species that can be referred to as *BRCA2*.

In the context of Wikification, people often train a ranking model to score how relevant a KB concept is to a mention. It is straightforward to use Wikipedia to supervise this model, since the hyperlink structure in Wikipedia text indicates which title a mention refers to. However, other KBs may not have such useful information. An entry in a typical biological KB only consists of a name, a few sentences of definition, synonyms, and a few relations (Figure 1). In addition, it is relatively difficult to obtain human annotations in the biomedical domain due to the high level of expertise required and to highly ambiguous concepts.

Our key contribution in this paper is to show that, by exploring the overlap and the relationship between KBs, we can obtain high quality indirect supervision signals for sufficiently many examples, and thus train a ranking model. Without using any document in training and no annotated supervision, our approach achieves better ranking results than all previous approaches tried on this problem.

We then explore another advantage of using multiple KBs; we show that, since concepts are represented in different ways in different KBs, there are some natural constraints between these representations. In the above example, if we determine that a gene mention is relevant to the human genome

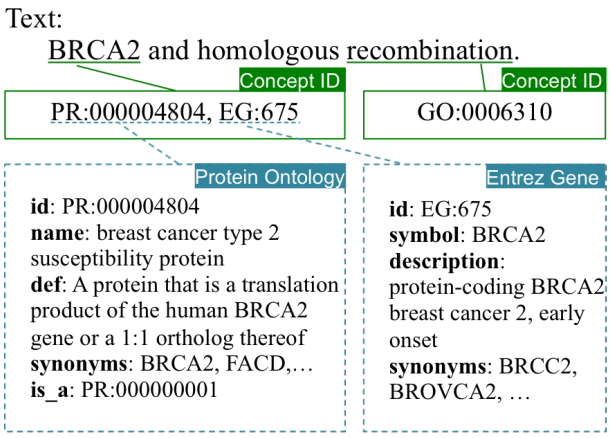


Figure 1: An example of concept annotations in the CRAFT dataset and of common attributes of concepts in the KBs.

and should therefore be grounded to human concepts in the NCBI Taxonomy, we can easily rule out all the candidate genes from other species, which are not mentioned in the document; we can develop these constraints since genes in the Entrez Gene KB have NCBI Taxonomy IDs as species attributes. If we do not use the NCBI Taxonomy as a knowledge source to ground concepts but rather only focus on disambiguating gene names in a document, we may lose this valuable information. Our final model combines this kind of prior knowledge with our ranker scores using a Constrained Conditional Model (CCM) (Roth and Yih, 2004; Chang et al., 2012) to enforce a coherent global mapping of all mentions in a given document to their corresponding concepts. The proposed system, CCMIS (CCM with Indirect Supervision), performs significantly better than the best unsupervised baseline and is competitive with a directly supervised model we use to assess the quality of the automatically generated indirect supervision.

2 Related Work

In the news domain, many researchers have studied ways to train a model to disambiguate concepts by directly using hyperlinks in Wikipedia documents as supervision. Earlier works (Bunescu and Pasca, 2006; Mihalcea and Csomai, 2007) focus on local features which compare context words with the content of candidate Wikipedia pages. Later, several

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/tacl_a_00089 by guest on 22 March 2026

works (Cucerzan, 2007; Milne and Witten, 2008; Han and Zhao, 2009; Ferragina and Scaiella, 2010; Ratinov et al., 2011) explore global features, trying to capture coherence among concepts that appear in close proximity in the text. Shen et al. (2012) and Dredze et al. (2010) train their model on a small manually created data set to handle documents in different domains. Cheng and Roth (2013) use relations between entities as constraints to support global inference with ranker scores, and show substantial improvement on several datasets. The main difference between our method and these Wikification approaches is that we train a ranking model by constructing indirect supervision signals from multiple KBs without using any annotated documents.

Concept Grounding and Word Sense Disambiguation (WSD) are closely related tasks as they both address the lexical ambiguity of language. Recently, several works try to relate the two by incorporating the lexical resources used in these tasks. Cholakov et al. (2014) disambiguate verbs to the senses in WordNet by creating semantic patterns from multiple lexical KBs, i.e., Wikipedia, Wikitionary, WordNet, FrameNet, and VerbNet, and also for each verb mention in the text. Moro et al. (2014) propose a graph-based approach which uses Wikipedia and WordNet as lexical resources. Their unified approach can achieve state of the art results on 6 Wikification and WSD datasets. The observation from these two papers are consistent with our conclusion that using multiple KBs jointly can improve individual tasks. Matuschek and Gurevych (2014) try to align different lexical resources (WordNet, Wikitionary, and Wikipedia in different languages). This approach is related to our construction of indirect supervision, and it would be interesting to see if the alignments could improve the quality of the indirect supervision and thus the quality of the disambiguation.

In the biomedical domain, the extensively studied word sense disambiguation problem (Weeber et al., 2001) focuses on disambiguating mentions to UMLS (Unified Medical Language System) Metathesaurus (Bodenreider, 2004). The main difference from our problem is that the WSD problem only addresses a small number of terms and the candidate concepts for each ambiguous mention are provided as part of the input. Researchers have developed various unsupervised methods that make use of information in the KB. McInnes (2008) compared the context words of the ambiguous mention to a profile built from UMLS concepts. Viewing the KB as a graph and adding context information into the graph, Agirre et al. (2010) compared the original PageRank algorithm with a personalized version. Jimeno-Yepes and Aronson (2010) automatically built training examples for each sense by retrieving documents from a large corpus. This approach is infeasible for our problem because we have a large amount of candidate concepts. The popular system MetaMap (Aronson and Lang, 2010) disambiguates mentions to semantic categories in UMLS using journal descriptor indexing. It is designed specifically for UMLS and it does not disambiguate two candidates if they are classified into the same semantic category. However, Jimeno-Yepes and Aronson (2010) showed that most of the unsupervised methods cannot even outperform the maximum frequency baseline and are not as good as the supervised methods (Joshi et al., 2005; Leroy and Rindflesch, 2005).

Recently, there has been a series of BioCreative challenges on gene normalization (Morgan et al., 2008; Lu and Wilbur, 2010; Mao et al., 2013) and chemical document indexing (Krallinger et al., 2013). These tasks are closer to the problem of automatic indexing of biomedical literature, however, all these studies focus on a single KB or even a subset of it.

In our experiments we make use of the CRAFT dataset that has been studied extensively; however, most of these studies focus on mention extraction rather than disambiguating mentions. Funk et al. (2014) comprehensively compared three dictionary-based systems: MetaMap, NCBO Annotator (Jonquet et al., 2009), and ConceptMapper (Tanenblatt et al., 2010) and shows that the latter has the best performance. However, it only applies various string matching strategies on the surface string of the mention and the concept names in KBs, and does not attempt any disambiguation based on the context of the mentions.

3 Task Definition and Model Overview

We formalize the problem as follows. We are given a document d with a set of mentions $M =$

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/tacl_a_00089 by guest on 22 March 2026

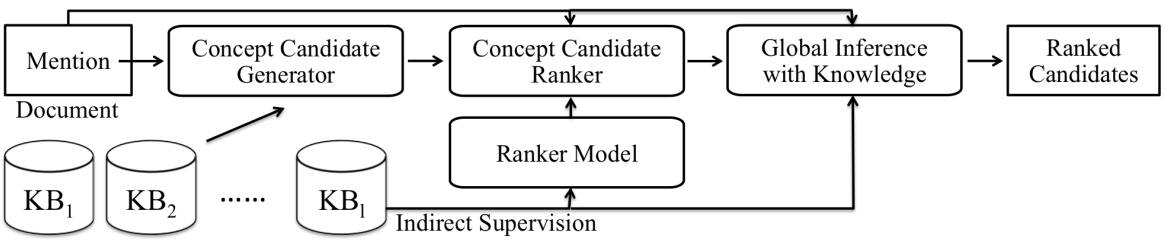


Figure 2: Algorithmic components of our system.

$\{m_1, \ldots, m_n\}$, and l KBs, $k_1, \ldots, k_l$. Each KB k_j is a graph (T_j, R_j), where a concept $t \in T_j$ represents a node and a relation $r \in R_j$ between two concepts is an undirected edge. For each mention in the document, our goal is to retrieve a set of concepts $\mathcal{C} \subseteq T_1 \cup \cdots \cup T_l$ that the mention refers to. Note that a mention may refer to multiple concepts in a single or multiple KBs.

Figure 2 shows the algorithmic components in our system. The first step is concept candidates generation. Given a mention m_i from a document, it produces a candidate set $C_{m_i} \subseteq T_1 \cup \cdots \cup T_l$. That is, C_{m_i} is a subset of all concepts in the KBs. We only look at the surface string of the mention in this step, no contextual information is examined. The goal of this step is to quickly produce a short list of concepts which includes the correct answers. We call these concept candidates "grounding candidates" for mention m_i.

The second and key step is the ranking step where, given a concept mention in text, we assign a score to each of its potential KB grounding candidates, which indicates how relevant it is to the given mention. We train a linear ranking SVM model using the information in the KBs. Note that unlike the Wikification problem in which it is possible to use the Wikipedia structure to learn a ranker, most other KBs do not have text with hyperlinks. To overcome this problem, we propose a novel method to utilize the redundancy and relationship between KBs as indirect supervision. Specifically, if two concepts in different KBs are determined to be the same, we can assume that one is the "gold label" for the other, and extract textual and relational features between them, making this pair an approximation of the real grounding instance. This method only requires information from the KBs hence no annotated document is needed.

We would like to use multiple sources of knowledge in order to train a robust ranking function over a set of candidates. Some of these can be captured via features of the ranking function (e.g., textual similarity between the context of a mention and the description of a concept in a KB); some, are better captured as constraints over the ranking produced by a ranking function. For example, if we choose to map a mention into a node in the KB that is species-specific, we insist on the species being mentioned in the context of the target mention. We will not link otherwise. As a way to combine statistical information and such declarative constraints we formulate our problem using Constrained Conditional Model (CCM) (Roth and Yih, 2004; Chang et al., 2012). We formulate the following Integer Linear Program (ILP) objective function to enforce a coherent global solution of all mentions in a document:

$$\arg\max_{\mathbf{e}} \quad \sum_{i=1}^{n} \sum_{j=1}^{|C_{m_i}|} s_i^j e_i^j - \sum_k \rho_k \gamma_k(\mathbf{e}) \qquad (1)$$
$$s.t. \quad e_i^j \in \{0,1\}, \quad \forall i, j$$

where e_i^j is a boolean variable that indicates if we choose the j-th candidate of the i-th mention, $\mathbf{e}$ is a vector that contains all the e_i^j's, and s_i^j is the ranker score, capturing the relatedness of mention m_i and its j-th candidate. In the second component, γ_k is a boolean variable that indicates whether the k-th constraint is violated, and ρ_k is the pre-defined penalty for violating the k-th constraint. Constraints are often defined on a subset of variables from our prior knowledge. For example, one constraint used in our system states that genes of a species can only be selected if that species is mentioned somewhere in the document. This ILP problem can be solved quickly by an off-the-shelf ILP package since only a small number of variables are constrained. In the end, all

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/tacl_a_00089 by guest on 22 March 2026

the selected concept candidates are ranked according to s_i^j, and a list of ranked concepts is returned for each mention.

In the following sections, we describe each component in detail.

4 Concept Candidate Generation

In the first step of our system, given a mention m from a document, we produce a set of concept candidates C_m which is a subset of all concepts in the KBs. We want to reduce the number of concept candidates from millions to a manageable size, so that a more sophisticated and resource-hungry algorithm can be applied to disambiguate them. There is a trade-off here: we want C_m to contain all the answers that correspond to m, but if there are too many candidates, the performance of the ranker may suffer. Therefore, we first do synonym matching, which gives a high precision candidate list. Word matching is then applied only if synonym matching fails to generate any candidate. We describe the synonym matching and word matching procedures in the following sections.

4.1 Synonym Matching

We construct a dictionary from all synonyms and name fields across all KBs. This dictionary maps a string (synonym or name) to all possible concepts. In order to handle variations of words, we use the SPECIALIST Lexical Tools [1] to normalize each token of synonyms and the given input mention. Doing exact matching between the mention and all the names in KBs gives a high precision candidate list.

4.2 Word Matching

After synonym matching, many mentions may still have an empty candidate list because the KBs do not cover all possible ways to express a concept. If no candidate is generated after applying the first dictionary lookup method, we compare words in the mention with words in the KBs and their synonyms. We use as candidates all those concepts that match in this process. Note that this strategy may return a large number of concepts, therefore we only keep the top k concepts to maintain feasibility. We use the score from the PageRank algorithm (Section 8.4) to rank concepts initially.

[1] http://lexsrv3.nlm.nih.gov/LexSysGroup/Projects/lvg/current/web/index.html

5 Concept Candidate Ranking

This section describes how we obtain the relevance scores s_i^j in Eq. (1) for each (mention, concept) pair. Given a mention m and a concept candidate $c \in C_m$, we define the *relevance* of c to m as:

$$s(m, c) = \phi(m, c) + \psi(c, \Gamma_m). \quad (2)$$

The first component $\phi(m, c)$ measures the local compatibility between the mention and the concept candidate. It uses text-based features to capture the intuition that a given concept c is more likely to be referred to by the mention m if the entry of c in a KB has high textual similarity to the text around m. We model it as a linear combination of a set of local features ϕ_i:

$$\phi(m, c) = \sum_i w_i \phi_i(m, c).$$

The second component of the scoring function (2), $\psi(c, \Gamma_m)$ is a global component that captures how well does the concept c fit into the disambiguation context Γ_m of the mention m. The disambiguation context consists of other concepts in the document or close to the mention. Of course, we do not know what the concepts that correspond to other mentions in the document are, and different ways to construct disambiguation context have been proposed (Cucerzan, 2007; Milne and Witten, 2008; Ratinov et al., 2011). Since in our case (in difference from the standard Wikification) a mention may refer to multiple concepts, using the current top ranked concept candidate from other mentions (Ratinov et al., 2011) may lose some useful information. Therefore, we develop an approach that is similar to Cucerzan (2007). Instead of considering all the ambiguous mentions in the document, we take all concept candidates from mentions in nearby sentences as our disambiguation context. Although some irrelevant concepts are included, we rely on a high precision candidate generation process to reduce errors. Similar to the local score model, we design a set of global features ψ_j across multiple KBs and define:

$$\psi(m, c) = \sum_j w_j \psi_j(c, \Gamma_m).$$

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/tacl_a_00089 by guest on 22 March 2026

In addition to local and global features, we use the PageRank score of the concept candidates as a baseline feature. To rank concept candidates $c \in C_m$ of a mention m, we use a linear ranking SVM to learn the weights w_i and w_j of the local and global features, respectively. The features used in our experiments are listed in the following sections.

5.1 Local Features

The local features used in our system are calculated from $context(m)$ and $def(c)$, where $context(m)$ represents the bag of words from p sentences before and after the mention m, and $def(c)$ is the bag of words from the definition of c in a KB. Words are lowercased and stemmed.

- $|context(m) \cap def(c)|$. The total number of common words in the context of m and c.
- Cosine similarity between the tf-idf vectors of $context(m)$ and $def(c)$. The i-th component in the vector is the tf-idf score of the i-th word in the vocabulary. The document frequency of words is calculated from all definitions in KBs, each definition representing a document.
- Common words in $context(m)$ and $def(c)$. This is a sparse boolean vector with length that is the size of vocabulary. The i-th feature is on if the i-th word in the vocabulary exists in both $context(m)$ and $def(c)$. Instead of using tf-idf vectors to capture the importance of each word, we use this feature to learn a weight for each word.

5.2 Global Features

Global features are defined on $neighbor(c)$ and Γ_m, where $neighbor(c)$ is the set of concepts which have relations with c in any of the KBs, and Γ_m is a set of candidate concepts from other mentions in the context of mention m. We consider all the mentions in p sentences before and after m in our experiments.

- $|neighbor(c) \cap \Gamma_m|$. The total number of common concepts in $neighbor(c)$ and Γ_m. We also split this number according to different KBs, and keep a feature that indicates the total number of common concepts from each KB.
- Common concepts in $neighbor(c)$ and Γ_m. This is a sparse boolean vector with length that is the total number of concepts. The i-th feature is on if the i-th concept exists in both $neightbor(c)$ and Γ_m.

6 Indirect Supervision

One of our key contributions in this paper is a way to train the model described above, without any supervision and no information (such as hypelinks) from the documents. To accomplish that, we devise an indirect supervision method that explores the redundancy of information in the KBs and the relationship between KBs to construct training examples, so that we can train a ranking SVM model without any annotated document.

We make the assumption that if two concepts from different KBs have the same cross reference field, they are, in fact, the same concept. For instance, the concept named *chromosome* is in the Gene Ontology (GO:0005694) and the Sequence Ontology (SO:0000340). These two entries both have an attribute "xref: Wikipedia:Chromosome"[2], which points to the Wikipedia page of chromosome thus indicating that they are the same concept. This redundancy allows us to generate an "annotated" example as follows: we make the definition of GO:0005694 the context of an ambiguous mention, and annotate it as the concept SO:0000340. This way, we can exploit the fact that definitions (of concepts) and related concepts are described differently in different KBs, to learn the importance of words and neighboring concepts, facilitating generalization. Another resource that we leverage is the "has participant" relationship. For example, *fructose metabolic process* (GO:0006000) in the Gene Ontology has a participant *fructose* (CHEBI:28757) from the Chemical Entities ontology. This allows us to generate another "annotated" example, where we annotate the *fructose* in *fructose metabolic process* with the concept CHEBI:28757. Note that while this information usually exists across multiple KBs, it is also possible to apply this method on a single KB.

Next we describe the indirect supervision process in some more details. The first step of constructing our training examples is to cluster concepts in the KBs by the cross reference attributes and also extract all pairs of concepts that have "has participant" relations. In each concept cluster, we randomly pick

[2]Besides Wikipedia, other knowledge bases can also be in the cross-reference field.

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/tacl_a_00089 by guest on 22 March 2026

one concept as the fake "mention" and the rest of concepts as the gold annotations to this mention.

6.1 Negative Concept Candidates

After the clustering step, we have obtained several positive concept candidates. To generate negative candidates, we apply our candidate generation method on the name of the concept which is treated as the mention, and also uniformly sample 200 concepts from all KBs. However, there is no guarantee that these candidates are really negative. Instead of using binary relevance score to train a linear ranking SVM, we take the number of common ancestors between a candidate and the positive candidates as the relevance score for the candidate. This way, if we missed a gold concept in the cluster, we won't assign it a completely irrelevant score if it has close proximity in the hierarchy of KB with other golds.

6.2 Feature Extraction

The local and global features we designed to capture the relatedness between a concept candidate and a mention are defined on $context(m)$ and Γ_m, which are the textual clues around the ambiguous mention m. The indirect supervision examples we have are not from any document, so there is no contextual clues. To approximate the features used at prediction time for the concept m which is treated as the mention, we use $def(m)$ to replace $context(m)$ and Γ_m is replaced by $neighbor(m)$, the neighboring concepts in the KB. By doing these approximations, we can generate features for a pair of concepts to facilitate training a ranker.

7 Constraints for Global Inference

At this point, we only use two types of hard constraints in Eq. (1) to enforce the consistency between concepts of different mentions. More specifically, a gene can be selected only if it is from a species mentioned somewhere in the document. We first form a species candidate set by gathering all concept candidates from NCBI Taxonomy[3] in a document. The assumption is that the genes mentioned in this document should be from at least one of the species in the species candidate set. Some concepts from the Protein Ontology and all concepts from the Entrez Gene Database have attributes that indicate the corresponding species, thus we design the following two constraints:

- A concept from the Entrez Gene Database must have an NCBI Taxonomy ID in the species candidate set of the document.
- If a concept from the Protein Ontology and a concept from NCBI Taxonomy have a relation "only in taxon", the concept from the Protein Ontology will be picked only if the concept from NCBI Taxonomy is in the species candidate set.

These two constraints are defined only on a single concept candidate, and we set the penalty ρ_k of them to be infinity to make these constraints hard constraints. Therefore, if a concept violates any of these two constraints, it will be excluded from the final concept list.

8 Evaluation

In this section, we compare the proposed CCMIS with five other approaches on the CRAFT dataset. In addition, we present experimental analysis designed to evaluate the candidate generation method, features of the ranking model, the quality of indirect supervision, and the benefit of using multiple KBs.

8.1 Dataset

The Colorado Richly Annotated Full-Text (CRAFT) corpus (Bada et al., 2012) is the largest gold standard corpus with high-quality annotations from multiple KBs: the Cell Type Ontology (CL), the Chemical Entities of Biological Interest ontology (CHEBI), the NCBI Taxonomy (NCBITaxon), the Protein Ontology (PR), the Sequence Ontology (SO), the Entrez Gene database (EG), and the Gene Ontology (GO). It identifies nearly all concepts from 67 full text of biomedical journal articles. We use the ontologies released along with the annotated documents in CRAFT-1.0 except EG which is not included in the package. We use the version which was available on October 30th, 2014. The CRAFT corpus consists of 82,634 concept mentions in total. The total number of concepts and unique concepts from each ontology is shown in Table 1. Note that

[3]NCBI Taxonomy Database is a curated classification and nomenclature for all of the organisms in the public sequence databases (`http://www.ncbi.nlm.nih.gov/taxonomy`)

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/tacl_a_00089 by guest on 22 March 2026

Ontology	#Concepts	#Anno.	#Uniq. Anno.
PR	26,879	15,593	889
NCBITaxon	789,509	7,449	149
GO	25,471	29,443	1,235
CHEBI	19,633	8,137	553
EG	17,097,474	12,266	1,021
SO	1,704	21,284	259
CL	857	5,760	155
Total	17,961,527	99,932	4,261

Table 1: Statistics of the concepts in the ontologies and the CRAFT corpus. We use "concepts" to refer to the entires in the ontologies, and "annotations" are concepts which are associated with mentions in the text. The second column shows the total number of concepts in each ontology. The third and fourth columns show the number of annotations and unique annotations of each ontology in the CRAFT corpus.

the total number of gold annotations (99,932 the last row of the third column) is larger than the number of mentions which indicates that a mention may refer to more than one concepts across multiple ontologies. The interannotator-agreement of concept annotations is above 90% F1 score for all the ontologies (Bada et al., 2012).

8.2 Evaluation Metrics

We mainly use the mean area under the precision-recall curve (AUC of PR-curve) (Agarwal et al., 2005) as the evaluation metric. Each mention has a ranked concept list as an output, and a set of gold concepts. We calculate the precision and recall at every ranking position, forming a PR-curve. Note that the recall is calculated using the total number of gold concepts, not just the total number of golds in the output list. This way we ensure this metric reflects the fact that some gold concepts are missing in the output. The AUC of the PR-curves of all mentions are averaged to get a final single number. We also report a hierarchical version of the AUC. The intuition is that if a concept is the parent or child of the gold concept, it should be penalized less than a concept which is far away from the gold in the hierarchy. We calculate hierarchical precision and recall using the method proposed in Kiritchenko et al. (2005), which replaces each concept by its ancestors (including itself), and then calculates the precision and recall at every ranking position by matching the ancestors of a concept candidate with the ancestors of the gold concepts. If a predicted concept has more common ancestors with the gold concepts, the score will be higher.

8.3 Baselines

We compare our proposed method with the following unsupervised methods.

- **TF-IDF** Cosine similarity between the TF-IDF vectors of mention context and the concept candidate's definition.
- **PageRank** (Brin and Page, 1998) We run the PageRank algorithm on the graph constructed from all the KBs with damping factor 0.85. This method doesn't consider any context of the ambiguous mentions at all, so a candidate concept always gets the same score, regardless of the mention it is a candidate for.
- **CollectiveInf** (Zheng et al., 2015) In this method, the initial score for each concept is calculated by a modified PageRank algorithm, in which the entropy of relations are used as the edges' weights. The final score of a concept candidate is further adjusted by the matching between neighboring concepts in the KB and the concept candidates around the mention. That is, if a neighbor concept in the KBs also appears in the context of the mention, the score of the concept candidate is increased according to the initial score of the matched neighbor concept.
- **Ppr** (Agirre and Soroa, 2009) The Personalized PageRank algorithm implemented in the UKB package[4]. This method first inserts the context mentions into the graph as nodes, and links them with directed edges to the corresponding concept candidates. The PageRank algorithm is then applied by concentrating the initial probability mass uniformly over the mention nodes. We take a window of 30 mentions as the context. Note that in order to have a fair comparison of the disambiguation ability, we use the proposed candidate generation method in Section 4 to produce the confusion set for each

[4] http://ixa2.si.ehu.es/ukb/

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/tacl_a_00089 by guest on 22 March 2026

Approach	Mean AUC	Mean hAUC
TF-IDF	40.44	48.50
PageRank	42.78	50.04
CollectiveInf	35.67	42.93
Ppr	43.39	51.88
Ppr_w2w	46.51	55.46
CCMIS	**48.58**	**57.37**

Table 2: A comparison of CCMIS and other five unsupervised approaches on the CRAFT corpus. The evaluation metrics are mean AUC of PR-curve and its hierarchical version. CCMIS outperforms other methods significantly in both metrics (using bootstrapped t-test with p-value < 0.05

Feature	Mean AUC	Mean hAUC
PageRank	42.78	50.04
+ Local Features	45.58	54.53
+ Global Features	46.64	56.00
+ Constraints	48.58	57.37

Table 3: Feature ablation study of the proposed method, CCMIS. The initial ranking of candidates is according to the PageRank score. Training an indirectly supervised ranker with local and global features improves the performance by 3.3 points of mean AUC. Doing global inference with constraints improves almost 2 points overall.

mention.

- **Ppr_w2w** This is another variant of the Personalized PageRank algorithm and it has the best performance in Agirre and Soroa (2009). It builds a graph for each target mention and concentrates the initial probability mass in the concept candidates of the target mention. We also directly use the implementation released in the UKB package and take 30 mentions around the target mention as the context. As discussed in Agirre and Soroa (2009), the drawback of this method is its slow running time, since it performs a PageRank algorithm on the whole graph for each mention. Given the large number of mentions and the huge graph in the CRAFT dataset, we set the number of iteration of PageRank to 3 in order to make the running time tractable. It takes around 3 days on a machine with 3.0GHz CPU, whereas other approaches only need less than one hour.

8.4 Experimental Results

We use a public linear ranking SVM package (Lee and Lin, 2014) with default parameters to learn the ranking model. Feature engineering is done by doing cross validation on the indirect supervision examples, therefore, we can use all documents as the test set for all approaches. Table 2 shows the overall performance of each approach. The results of graph-based approaches are consistent with the results in Agirre and Soroa (2009): Ppr_w2w performs better than Ppr, and these two Personalized PageRank approaches outperform the static PageRank method. However, given the large size of KBs and the number of mentions in the CRAFT corpus, Ppr_w2w requires two days to run on a 3.0GHz CPU, whereas Ppr only takes two hours and other methods can be done within an hour. TF-IDF does not performs well since the definitions of concepts are very short and concise, which makes them hard to be matched with any context words. Our algorithm CCMIS gets 2 points higher than Ppr_w2w in terms of mean AUC, even though no additional external information is being used; specifically, no annotated document is needed to train the ranking model. Regarding the relaxed metric, mean hierarchical AUC (hAUC), the relative performance is the same but the gaps between hAUC and AUC indicate that many concept candidates are the ancestors or descendants of the gold concept, which might be proven good enough in practice.

Table 3 shows a feature ablation study of CCMIS. The initial ranking of candidates is according to the static PageRank score. Training an indirectly supervised ranker with local features adds almost 3 points of mean AUC. Without adding the two constraints to enforce species coherence, the ranking scores from our ranker already perform better than other approaches. Using these constraints adds about 1.9 points of mean AUC overall.

The candidate generation method plays an important role in getting good ranking performance. In CCMIS, we include the top 10 candidates from word matching only when synonym matching fails to generate any candidate since candidates generated by word matching are nosier. This way covers 68.11% of the gold concepts, which indicates that the ceiling of the ranking performance is close to 68.11. To

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/tacl_a_00089 by guest on 22 March 2026

Approach	$k = 0$	$k = 10$	$k = 20$	$k = 30$
TF-IDF	38.60	21.30	17.58	15.12
PageRank	40.09	21.78	20.23	19.73
CollectiveInf	33.93	16.6	12.74	11.17
Ppr	40.91	20.71	20.40	19.74
Ppr_w2w	42.58	24.56	23.13	21.21
CCMIS	**45.95**	**29.32**	**26.46**	**24.48**
Gold coverage	62.70	68.92	70.02	70.62

Table 4: Comparing ranking performance by changing the parameter in the candidate generation algorithm. Besides synonym matching, we use word matching to make sure each mention has at least k candidates. Note that in the setting of Table 2, word matching is only applied if synonym matching fails to generate any candidates. The gold coverage is the percentage of gold annotations included in the candidate list, a performance upper bound. The metric is mean AUC.

Approach	Mean AUC	Mean hAUC
CCMIS	48.58	57.34
Gold Clusters	50.85	56.50
Direct Supervision	58.98	62.59

Table 5: Evaluating the quality of indirectly supervised examples. The only difference between these three approaches is the way we obtain training examples. That is, only the ranking model is changed. Concept candidates, features, and learning algorithm are stay the same.

show how the candidate generation method affects ranking performance we add candidates from word matching to mentions so that each mention has at least k concept candidates. The results are shown in Table 4. The last row of Table 4 shows the percentage of gold concepts included as candidates. We can see that after $k = 20$, the gold coverage merely increases. This indicates that lexical level matching is not sufficient for generating more gold concepts into the candidate set. From $k = 0$ to 10, the performance of each approach drops a lot. CCMIS is more robust when there are more irrelevant candidates. It is also interesting to see that simply increasing the gold coverage may result in worse overall performance. We need a more powerful ranking algorithm to handle larger number of candidates.

8.5 Quality of Indirect Supervision

We assess the quality of our indirect supervision training examples by comparing CCMIS's performance with two other approaches. These two approaches only change the way CCMIS constructs training examples for linear ranking SVM. That is, only the ranking model is changed while other components (concept candidates, features, and learning algorithm) of the system are identical.

Instead of finding concept clusters using cross reference fields and has_participant relations as in our proposed method, the first approach used the gold clusters from the mentions which have more than one gold annotation in the CRAFT corpus. Each mention forms a concept cluster in which members are the gold annotations. We conduct 5-fold cross validation on the CRAFT corpus, where gold clusters are extracted from the training documents. Note that the features of the training examples are generated in the same way as in our indirect supervision method, that is, although concept clusters are taken from documents, no text is used to generate features. This way we can focus on comparing the quality of the concept clusters obtained from the KBs with human annotation. This approach is named Gold Clusters in Table 5. Interestingly, its performance is better than CCMIS in terms of mean AUC but slightly worse in mean hAUC, which indicates that the concept clusters obtained by the proposed method have as good a quality as the gold annotations.

The second comparison is against direct supervision; here we use gold annotations from the CRAFT corpus itself and generate features of training examples in the same way used at prediction time. The results of 5-fold cross validation are listed in the third row of Table 5. Apparently, but not surprisingly, training with gold achieves about 10 points better than CCMIS. Note that in this case, the test examples, which are generated given the text documents, are expected to be more similar to the training examples, which are also generated from the text documents, in difference from the training examples used by CCMIS. This gap indicates how well the indirect supervision method approximates the distribution of the features in the test data, without using any document to obtain a good model.

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/tacl_a_00089 by guest on 22 March 2026

Approach	Individual KBs							Joint Approach (on individual KBs)						
	PR	GO	NC	EG	CH	SO	CL	PR	GO	NC	EG	CH	SO	CL
PageRank	83.1	36.9	44.7	11.6	56.4	57.9	74.3	83.4	41.7	45.2	37.9	71.4	56.9	77.8
CollectInf	83.4	35.7	44.7	11.6	56.3	57.7	74.2	84.0	39.2	45.0	45.5	60.5	57.2	80.5
Ppr	83.2	36.5	44.7	11.6	58.0	57.8	76.1	82.9	41.3	45.1	53.9	71.2	57.2	78.9
Ppr_w2w	84.5	35.7	44.3	23.2	70.6	56.9	76.6	84.5	40.3	44.9	32.3	71.2	57.5	77.6
CCMIS	84.4	35.5	43.7	25.9	68.6	57.0	76.5	83.9	42.3	45.1	38.5	70.7	57.7	78.0

Table 6: A comparison between linking to each ontology individually and jointly. The evaluation metric is mean AUC of PR-curve. Note that the numbers are not directly comparable with the ones in the previous tables since the mentions in the CRAFT corpus are split into different datasets according to the annotations. For each approach and ontology, jointly using multiple KBs yields better results in most cases. The averaged performance over all datasets is summarized in Table 7.

8.6 Using KBs Individually v.s. Jointly

We compare the ranking performance of using KBs individually versus jointly. The joint case is exactly the setup of our task: grounding a given mention to multiple KBs, where the information from multiple KBs is used together. In the individual case, we only use the information from a single KB to ground concepts to this target KB, and do this for each KB. We create a dataset for each KB by splitting the annotations in the CRAFT dataset. For example, when we create the dataset for the Gene Ontology (GO), we only keep the mentions that have at least one gold annotation from GO, and remove all the other annotations. Approaches applied on this dataset can only access the information in GO. Note that we keep the candidate generation process the same as what we do in the joint case, but only keep the concept candidates from the target ontology. Hence, we can see how does ranking performance changes given the same set of candidates. The evaluation is done on each ontology's dataset separately. We also evaluate the joint methods on each ontology separately by splitting the final ranked concepts according to their knowledge source. It allows for a fair comparison of the performance of the joint method with the individual methods. Note that the performance numbers in this section are not directly comparable to the ones in previous sections as the dataset has changed.

The results of applying different approaches to each KB's dataset are shown in Table 6. We can see that for each pair of (approach, ontology), using multiple KBs jointly usually yields a better result than using each KB individually. The improvement is more obvious for some ontologies, for instance, EG and CH, which contain sparse relations. In these cases, using multiple KBs together provides more information of the neighbor concepts thus may have a better match with the context of the ambiguous mention.

Table 7 shows the overall performance by averaging the AUC score of each mention in each ontology's dataset. CCMIS outperforms other approaches in the joint case, but has the largest gap between the individual and joint cases. The reason is that the concept clusters used to generate our training examples have worse quality and quantity within a single KB. In addition, using a single KB makes the global features sparser. This result indicates that CCMIS leverages the information across multiple knowledge bases well to achieve the best overall performance. Interestingly, Ppr_w2w achieves the best performance in the individual case. It seems that approach performs relatively well in a homogeneous network. It would be interesting to see if we can combine the power of Ppr_w2w as a feature in our ranking model (while avoiding its unrealistic computational cost).

9 Discussion and Conclusions

This work studied the concept grounding problem where the target knowledge bases do not contain rich textual and structural information. We showed that we can achieve better performance than existing methods by leveraging the relations between multiple KBs. The proposed approach of constructing indirect supervision examples enables us to apply the well-studied statistical learning model even

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/tacl_a_00089 by guest on 22 March 2026

Approach	Individual KBs	Joint Approach
PageRank	49.85	55.74
CollectiveInf	49.46	55.42
Ppr	52.12	54.88
Ppr_w2w	**52.23**	56.18
CCMIS	49.93	**57.65**

Table 7: The overall performance of using KBs individually and jointly. Note that the numbers are averaged AUC of mentions across different KBs' datasets, a different evaluation metric from Table 2. Using multiple KBs jointly always yields a better result and the gain of CCMIS is the largest.

when there is no direct supervision. Inducing simple constraints to enforce solution consistency across related KBs was shown to further improve the ranking results. This work and the analysis shown suggest a range of questions from how to combine other resources to obtain higher quality of supervision, to issues of handling feature sparsity and improving the crucially important candidate generation precision.

An immediate question that follows from our work is whether (and what) other tasks can be benefit from the proposed technique. The proposed method of constructing indirect supervision examples is based on (1) Redundant information between multiple knowledge bases. The fact that duplicated concepts with different descriptions/relations appear in different KBs allows the algorithm to figure out what is important in the concept descriptions and thus provides a way to distinguish among concepts. (2) The features extracted from concept-concept pairs. These, as we show, approximate well the features of mention-concept pairs at test time. If (1) is satisfied, that is, there are multiple KBs and enough entries in them that can be aligned, then the proposed method can be applied. However, the performance of this method highly depends on (2), the quality of features and how well the indirect supervision examples approximate the text at prediction time.

An application which fits this setting is the verb sense disambiguation problem, where there are multiple sense inventories (e.g., VerbNet (Kipper et al., 2000), FrameNet (Baker et al., 1998), and PropBank (Palmer et al., 2005)) and many of the senses are aligned by different resources (e.g., UBY (Gurevych et al., 2012) and Unified Verb Index[5]). There are corpora which have annotations from one or more these verb sense inventories available, such as OntoNotes (Pradhan et al., 2007) and MASC[6]. However, unlike the biomedical ontologies which have many common attributes and relatively uniform structure, different verb sense inventories vary in format and content: some resources have descriptions or example sentences of the senses, but others only have the names of semantic roles; some have relations between senses but some do not. Therefore, the question of what features would be useful in this case could be very different from those proposed in this paper and would require additional research.

In one of the related works we mention in Section 2, Cholakov et al. (2014) actually utilized multiple verb sense inventories to link verbs to VerbNet. They generate a "semantic pattern" for each sense using the connections between different sense inventories, and those to each verb mention in the text. The prediction is based on the similarity between semantic patterns. Although this unsupervised method is very different from our indirect supervision approach, they confirm that using links between different sense inventories improves the performance. It would be very interesting to try our method on this problem.

Acknowledgments

This research is supported by NIH grant U54-GM114838, and a grant from the Allen Institute for Artificial Intelligence (allenai.org).

References

Shivani Agarwal, Thore Graepel, Ralf Herbrich, and Dan Roth. 2005. A large deviation bound for the area under the roc curve. In *The Conference on Advances in Neural Information Processing Systems (NIPS)*, pages 9–16.

Eneko Agirre and Aitor Soroa. 2009. Personalizing pagerank for word sense disambiguation. In *Proceedings of the 12th Conference of the European Chap-*

[5] http://verbs.colorado.edu/verb-index/index.php

[6] https://catalog.ldc.upenn.edu/LDC2013T12

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/tacl_a_00089 by guest on 22 March 2026

ter of the Association for Computational Linguistics, pages 33–41.

Eneko Agirre, Aitor Soroa, and Mark Stevenson. 2010. Graph-based word sense disambiguation of biomedical documents. *Bioinformatics*, 26(22):2889–2896.

Alan R. Aronson and François-Michel Lang. 2010. An overview of MetaMap: Historical perspective and recent advances. *Journal of the American Medical Informatics Association*, 17(3):229–236.

Michael Ashburner, Catherine A. Ball, Judith A. Blake, David Botstein, Heather Butler, J. Michael Cherry, Allan P. Davis, Kara Dolinski, Selina S. Dwight, Janan T. Eppig, Midori A. Harris, David P. Hill, Laurie Issel-Tarver, Andrew Kasarskis, Suzanna Lewis, John C. Matese, Joel E. Richardson, Martin Ringwald, Gerald M. Rubin, and Gavin Sherlock. 2000. Gene Ontology: Tool for the unification of biology. *Nature Genetics*, 25(1):25–29.

Michael Bada, Miriam Eckert, Donald Evans, Kristin Garcia, Krista Shipley, Dmitry Sitnikov, William A. Baumgartner, K. Bretonnel Cohen, Karin Verspoor, Judith A. Blake, and Lawrence E. Hunter. 2012. Concept annotation in the CRAFT corpus. *BMC Bioinformatics*, 13(1):161.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *Proceedings of the 17th International Conference on Computational Linguistics*, pages 86–90.

Olivier Bodenreider. 2004. The unified medical language system (UMLS): Integrating biomedical terminology. *Nucleic Acids Research*, 32:D267–D270.

Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1):107–117.

Razvan Bunescu and Marius Pasca. 2006. Using encyclopedic knowledge for named entity disambiguation. In *Proceedings of the European Chapter of the ACL (EACL)*.

Ming-Wei Chang, Lev Ratinov, and Dan Roth. 2012. Structured learning with constrained conditional models. *Machine Learning*, 88(3):399–431, 6.

Xiao Cheng and Dan Roth. 2013. Relational inference for wikification. In *Proceedings of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*.

Kostadin Cholakov, Judith Eckle-Kohler, and Iryna Gurevych. 2014. Automated verb sense labelling based on linked lexical resources. In *EACL*, pages 68–77.

Silviu Cucerzan. 2007. Large-scale named entity disambiguation based on Wikipedia data. In *Proceedings of EMNLP*, pages 708–716.

Mark Dredze, Paul McNamee, Delip Rao, Adam Gerber, and Tim Finin. 2010. Entity disambiguation for knowledge base population. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 277–285.

Paolo Ferragina and Ugo Scaiella. 2010. TAGME: On-the-fly annotation of short text fragments (by Wikipedia entities). In *Proceedings of CIKM*, pages 1625–1628.

Christopher Funk, William Baumgartner, Benjamin Garcia, Christophe Roeder, Michael Bada, K. Bretonnel Cohen, Lawrence E. Hunter, and Karin Verspoor. 2014. Large-scale biomedical concept recognition: An evaluation of current automatic annotators and their parameters. *BMC Bioinformatics*, 15(1):59.

Iryna Gurevych, Judith Eckle-Kohler, Silvana Hartmann, Michael Matuschek, Christian M. Meyer, and Christian Wirth. 2012. UBY: A large-scale unified lexical-semantic resource based on LMF. In *Proceedings of EACL*, pages 580–590.

Xianpei Han and Jun Zhao. 2009. Named entity disambiguation by leveraging Wikipedia semantic knowledge. In *Proceedings of CIKM*, pages 215–224.

Antonio J. Jimeno-Yepes and Alan R. Aronson. 2010. Knowledge-based biomedical word sense disambiguation: Comparison of approaches. *BMC Bioinformatics*, 11(1):569.

Clement Jonquet, Nigam Shah, and Mark Musen. 2009. The open biomedical annotator. *Summit on Translational Bioinformatics*, 2009:56.

Mahesh Joshi, Ted Pedersen, and Richard Maclin. 2005. A comparative study of support vector machines applied to the supervised word sense disambiguation problem in the medical domain. In *Proceedings of IICAI*, pages 3449–3468.

Karin Kipper, Hoa Trang Dang, and Martha Palmer. 2000. Class-based construction of a verb lexicon. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, Austin, TX. AAAI.

Svetlana Kiritchenko, Stan Matwin, and Fazel Famili. 2005. Functional annotation of genes using hierarchical text categorization. In *Proceedings of BioLINK SIG: Linking Literature, Information and Knowledge for Biology*.

Martin Krallinger, Florian Leitner, Obdulia Rabal, Miguel Vazquez, Julen Oyarzabal, and Alfonso Valencia. 2013. Overview of the chemical compound and drug name recognition (CHEMDNER) task. In *BioCreative Challenge Evaluation Workshop*, volume 2, page 2.

Ching-Pei Lee and Chih-Jen Lin. 2014. Large-scale linear RankSVM. *Neural computation*, 26(4):781–817.

Gondy Leroy and Thomas C. Rindflesch. 2005. Effects of information and machine learning algorithms on

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/tacl_a_00089 by guest on 22 March 2026

word sense disambiguation with small datasets. *International Journal of Medical Informatics*, 74(7):573–585.

Zhiyong Lu and W. John Wilbur. 2010. Overview of BioCreative III gene normalization. In *Proceedings of the BioCreative III Workshop*, pages 24–45.

Yuqing Mao, Kimberly V. Auken, Donghui Li, Cecilia N. Arighi, and Zhiyong Lu. 2013. The gene ontology task at BioCreative IV. In *Proceedings of the Fourth Biocreative Challenge Evaluation Workshop*, volume 1, pages 119–127.

Michael Matuschek and Iryna Gurevych. 2014. High performance word sense alignment by joint modeling of sense distance and gloss similarity. In *Proceedings of COLING*, pages 245–256.

Bridget T. McInnes. 2008. An unsupervised vector approach to biomedical term disambiguation: Integrating UMLS and Medline. In *Proceedings of ACL: Student Research Workshop*, pages 49–54.

Rada Mihalcea and Andras Csomai. 2007. Wikify!: Linking documents to encyclopedic knowledge. In *Proceedings of ACM Conference on Information and Knowledge Management (CIKM)*, pages 233–242.

David Milne and Ian H. Witten. 2008. Learning to link with Wikipedia. In *Proceedings of ACM Conference on Information and Knowledge Management (CIKM)*, pages 509–518.

Alexander A. Morgan, Zhiyong Lu, Xinglong Wang, Aaron M. Cohen, Juliane Fluck, Patrick Ruch, Anna Divoli, Katrin Fundel, Robert Leaman, Jörg Hakenberg, Chengjie Sun, Heng-Hui Liu, Rafael Torres, Michael Krauthammer, William W. Lau, Hongfang Liu, Chun-Nan Hsu, Martijn Schuemie, K. Bretonnel Cohen, and Lynette Hirschman. 2008. Overview of BioCreative II gene normalization. *Genome Biology*, 9:S3.

Andrea Moro, Alessandro Raganato, and Roberto Navigli. 2014. Entity linking meets word sense disambiguation: A unified approach. In *Transactions of the Association for Computational Linguistics*, volume 2, pages 231–244.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106, March.

Sameer S. Pradhan, Eduard Hovy, Mitch Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2007. Ontonotes: A unified relational semantic representation. *International Journal of Semantic Computing*, 01(04):405–419.

Lev Ratinov, Doug Downey, Mike Anderson, and Dan Roth. 2011. Local and global algorithms for disambiguation to Wikipedia. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Dan Roth and Wen-Tau Yih. 2004. A linear programming formulation for global inference in natural language tasks. In *Proceedings of the Annual Conference on Computational Natural Language Learning (CoNLL)*, pages 1–8. Association for Computational Linguistics.

Wei Shen, Jianyong Wang, Ping Luo, and Min Wang. 2012. LINDEN: Linking named entities with knowledge base via semantic knowledge. In *Proceedings of WWW*, pages 449–458.

Michael A. Tanenblatt, Anni Coden, and Igor L. Sominsky. 2010. The ConceptMapper approach to named entity recognition. In *Proceedings of LREC*.

Marc Weeber, James G. Mork, and Alan R. Aronson. 2001. Developing a test collection for biomedical word sense disambiguation. In *Proceedings of the AMIA Symposium*, page 746.

Jin G. Zheng, Daniel Howsmon, Boliang Zhang, Juergen Hahn, Deborah McGuinness, James Hendler, and Heng Ji. 2015. Entity linking for biomedical literature. *BMC Medical Informatics and Decision Making*, 15(Suppl 1):S4.

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/tacl_a_00089 by guest on 22 March 2026