Similarity-based Distant Supervision for Definition Retrieval

Jiepu Jiang
Center for Intelligent Information Retrieval,
College of Information and Computer Sciences,
University of Massachusetts Amherst
jpjiang@cs.umass.edu

James Allan
Center for Intelligent Information Retrieval,
College of Information and Computer Sciences,
University of Massachusetts Amherst
allan@cs.umass.edu

ABSTRACT

Recognizing definition sentences from free text corpora often requires hand-crafted patterns or explicitly labeled training instances. We present a distant supervision approach addressing this challenge without using explicitly labeled data. We use plausibly good but imperfect definition sentences from Wikipedia as references to annotate sentences in a target corpus based on text similarity measures such as ROUGE. Experimental results show our approach is highly effective, generating noisy but large, useful, and localized training instances. Definition sentence retrieval models trained using the synthesized training examples are more effective than those learned from manual judgments of a few thousand sentences. We also examine different text similarity measures for annotation, including both unsupervised and supervised ones. We show that our method can significantly benefit from supervised text similarity measures learned from either external training data (from the SemEval Semantic Text Similarity task) or local ones (a few hundred judged sentences on the target corpus). Our method offers a cheap, effective, and flexible solution to this task and can benefit a broad range of applications such as web search engines and QA systems.

KEYWORDS

Distant supervision; definition sentence retrieval; definitional question answering; semantic textual similarity.

1 INTRODUCTION

Definition helps readers quickly comprehend terms and concepts. Identifying *definition sentences*—sentences that interpret terms and concepts—from texts is useful to many applications such as hypernym extraction [35], automatic thesaurus construction [26], question answering [9, 39], and so on. Many search engines also display definition sentences as direct answers [2, 7], along with regular search results, if the query contains a technical term.

Early methods [25, 37] developed hand-crafted lexical-syntactic patterns such as "`X is (a) Y`" to recognize definition sentences. These patterns have limited effectiveness because it is difficult and time-consuming to enumerate all possible patterns and exceptions. State-of-the-art methods [9, 16, 26] rely on supervised machine learning techniques to identify definition sentences. These methods are more effective, but they also have two key limitations:

- **Annotation cost** – it typically requires at lest a few thousand labeled sentences to train effective supervised models.
- **Generalizability** – the labeled sentences and the learned models are often pertinent to a particular corpus, which may not generalize well to other corpora.

These limitations make it difficult to apply existing models and training data to effectively address problems in a *new* corpus. Particularly, we note that it is difficult to maintain the effectiveness of a supervised model even when the new corpus looks very similar to the old ones. For example, as reported in Section 6, definition sentence retrieval models learned from an existing corpus of ACL anthology[1] articles yield limited accuracy on a new corpus of ACM digital library articles, where the two corpora are similar to each other in terms of both style and topic.

To address these challenges, we propose a distant supervision method to generate large and effective definition sentence judgments on new text corpora requiring little manual annotation effort. Distant supervision [24] (DS) refers to a supervised learning paradigm where the training data is not manually annotated, but automatically generated from knowledge bases (KBs) and heuristics. We apply this paradigm to our task. We first extract a small set of accurate definition sentences from a KB—in our case, Wikipedia. Then, we use these sentences as references (examples) to automatically generate definition sentence judgments in a target corpus. Last, we train supervised models using such synthesized judgments and apply them to identify definition sentences in the target corpus. Figure 1 shows an example of the procedure.

Our approach has the following advantages:

- It generates large-scale and accurate training data requiring little human annotation effort, which can be applied to develop effective techniques for a *new* corpus instantly.
- It produces *localized* training instances pertinent to the target corpus of interest, ensuring that the learned models are representative of the corpus.
- Existing KBs such as Wikipedia cover many topics, making it easy to apply our method to corpora of different domains.
- Although the synthesized judgments are generated using examples from a KB, the learned model can accurately find definition sentences for terms that do not exist in the KB.

We call our method *similarity-based distant supervision* (sim-DS) because we annotate sentences in the target corpus based on their similarities to the KB's examples. As Figure 1 shows, we assign the second sentence a higher score than the first one because it is more similar to the example. Our method brings together previous

[1] http://aclweb.org/anthology/

CIKM'17, November 6–10, 2017, Singapore.
 ISBN 978-1-4503-4918-5/17/11…$15.00
DOI: https://doi.org/10.1145/3132847.3133032

Table 1: Criteria used for assessing definition sentences.

Quality Level	Example Sentence (underline indicates the target term)
3 (informative & dedicated)	ImageNet is an image dataset organized according to the WordNet hierarchy.
2 (informative & not dedicated)	We train the baseline models using the ImageNet dataset (including 1000 object classes and 1.4M images).
1 (only basic fact)	OSPF is a routing protocol.
0 (not explanatory)	This requires a parallel sorting of the subgraphs sizes (number of critical nodes).

techniques in three areas: distant supervision [24], reference-based evaluation techniques [20, 29], and semantic textual similarity [1]. However, our problem is also unique and challenging because:

First, most current DS methods annotate instances using simple rules such as matching word occurrences. For example, Snow et al. [35] used two words with known hypernym relation in WordNet to annotate their co-occurring sentences. In contrast, we tackle a more challenging annotation task, which requires an appropriate text similarity measure to generate judgments.

Second, reference-based evaluation [20, 29] had achieved great success in language generation tasks such as text summarization and machine translation, but previous studies mostly used human-created ground truths as references. In contrast, our method is less expensive but more challenging because we heuristically extract references from Wikipedia, which may not guarantee to be correct.

Third, our DS method can work in a supervised manner. It can be refined automatically with the help of some manual judgments. We use manual judgments to train supervised text similarity measures, which can generate more accurate DS judgments than using unsupervised similarity functions such as ROUGE. This offers an effective method to combine automatic and human judgments.

Experimental results show that our approach generates large-scale, localized, and effective definition sentence judgments. Models trained using our method outperform those using: 1) manual judgments of a few thousand sentences; 2) a large set of heuristic judgments automatically extracted from Wikipedia. The trained models are also more representative of the characteristics of the target corpus than other models.

2 RELATED WORK

2.1 Recognizing Definition from Texts

We focus on the task of definition sentence retrieval—giving a term or concept, we sort sentences in a text corpus by how well they interpret the term. This task is closely related to definition sentence classification [3, 16, 26] and definitional question answering (QA) [8, 9, 17]. Many previous studies [3, 10, 11, 16, 26] also further extract the term being defined and its definition from sentences, but we only focus on definition sentence retrieval here. The problem also shares similarities to hypernym extraction [12, 35], because sometimes a hypernym can explain its hyponym (although usually not expressive). Definition sentence retrieval is useful to many applications, such as improving QA systems, generating direct answers in web search engines, selecting candidate sentences to assist definition and hypernym extraction, and so on.

Most existing solutions rely on the context of a term to identify whether or not the sentence interprets the term. Early approaches [12, 25, 37] often rely on hand-crafted lexical-syntactic patterns such as "`is a/an`" and "`is defined as`". Recent methods are mostly based on supervised machine learning, which allows discovering these patterns from labeled data. For example: Cui, Kan, and Chua [9] learned n-gram models for definitional QA; Navigli and Velardi [26] learned a generalized representation of definition sentences based on word lattice; Boella and Di Caro extracted definition and hypernym relations using syntactic dependencies [3]; Jin, Kan, Ng, and He [16] modeled the task of definition extraction as a sequential tagging problem.

Despite being effective, supervised learning methods require a decent amount of training instances to work well. Existing datasets for this task [16, 26] mostly include a few thousand labeled sentences, which requires a substantial amount of human annotation effort. An effective way of getting free training data for this task is to consider the first sentence of a Wikipedia entry as a definition sentence. However, patterns learned directly from these Wikipedia sentences are usually influenced by the Wikipedia corpus. A few previous studies [10, 33] also applied semi-supervised methods to address this issue.

2.2 Distant Supervision

Our method differs from existing supervised approaches for this task in that we automatically generate training instances on a target corpus using Wikipedia. This is similar to distant supervision [24]—using knowledge bases and heuristics to annotate a target corpus. Previous studies have successfully applied distant supervision to many different natural language processing tasks, including hypernym extraction [35], relation extraction [24], open information extraction (IE) [38], named entity recognition [34], sentiment analysis [32], and so on.

An important step of distant supervision (DS) is to create an automatic annotation rule. Previous studies mostly relied on simple rules for annotation such as matching word occurrences [24, 35]. In contrast, we use text similarity measures to generate DS judgments—sentences that are sufficiently similar to existing definition sentences are automatically assessed as definitional sentences. This is also related to Intxaurrondo et al.'s [14] study, where they used a similar idea for event extraction in social media.

2.3 Reference-based Evaluation

Reference-based evaluation methods assess machine-generated results by comparing them to ground truth ones (usually human-generated) using text similarity measures. It has been widely applied to evaluate language generation tasks such as text summarization [13, 20, 21, 27]) and machine translation [29]. These methods mostly used the overlap of linguistic units (such as unigrams, bigrams, etc.) between two texts to measure their similarity to each other. A few recent studies also considered semantic matching [28, 30] to address vocabulary mismatch issues.

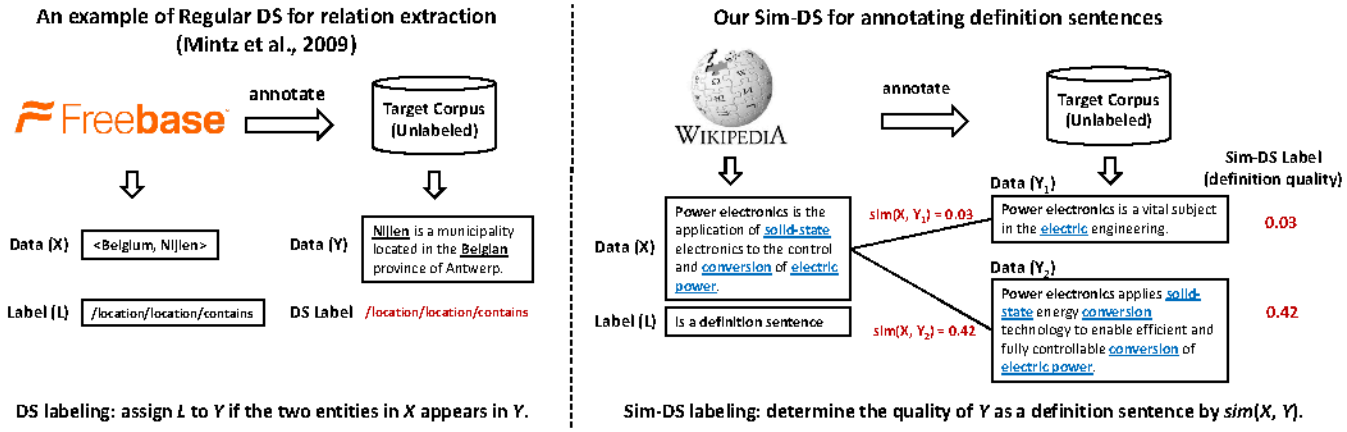

Figure 1: Examples of distant supervision for relation extraction (left) and similarity-based distant supervision for definition sentence retrieval (right).

In contrast to its popularity in language generation tasks, reference-based evaluation is less common in information retrieval. Xu et al. [39] applied ROUGE to evaluate definitional question answering, which is closely related to our task. Carterette and Allan [6] used cosine similarity to bootstrap relevance judgments to evaluate document retrieval. Our work is similar to these studies, but we use the synthesized judgments mainly for training purposes. Besides, our method also differs from previous studies in that we use imperfect references to assess other sentences (the first sentence of a Wikipedia entry is not always a good definition sentence), which makes the problem more challenging.

3 PROBLEM DEFINITION

We define a **definition sentence** as one that informs readers "*what is (an) X*", where X is a term[2]. It does not restrict to explicit definitive statements such as "*X is defined as …*", although we believe an effective solution should recognize such sentences as high-quality results. Table 1 shows some example definition sentences of different quality levels. The criteria are mainly to help search engines retrieve definition sentences as direct answers.

We define the task of **definition sentence retrieval** as: *given a target term X, retrieve and rank sentences in a text corpus by their quality of explaining X.* The input is a target term such as "*BM25*" or "*Barycentric Spanner*" and we aim to produce a ranking of sentences. Supervised learning techniques for this task require the following forms of training data: *<target term, sentence, score>*. Our goal is to automatically generate such training data on *a target corpus* based on a knowledge base and (optionally) a few manual judgments. We study the following questions:

- **RQ1** – How to generate effective training data for this task on a target corpus only based on a knowledge base?
- **RQ2** – Can we improve the accuracy of the automatically generated training data using a few manual judgments? Is it worthwhile to do so (compared to directly using the manual judgments for training)?

[2] Here a term refers to "a word or phrase used to describe a thing or to express a concept, especially in a particular kind of language or branch of study" (Oxford Dictionary), which does not mean index term in information retrieval and is not restricted to a single word.

4 SIMILARITY-BASED DISTANT SUPERVISION

4.1 Framework

Distant supervision [24] is a machine learning paradigm that uses knowledge bases (KBs) and heuristics to annotate a target corpus. We can summarize the procedure of existing DS methods (regular DS) as follows.

> **Regular DS**: given a data entry X with the label L (usually obtained from a KB and is assumed to be correct), assigns L to a data entry Y in the target corpus if X and Y satisfy a rule $R(X, Y)$.

Figure 1 (left) shows an example of the DS method [24] applied to relation extraction (the task is to infer the relationship of two entities from their co-occurrence sentences), where: X is a pair of entities <Belgium, Nijlen>; L is a known relationship in Freebase /location/location/contains assigned to this pair of entities; Y is a sentence from the target corpus; $R(X, Y)$ is satisfied iff X (the two entities) appears in Y (the sentence).

The annotation rule R in regular DS is usually straightforward and precise, such as matching word occurrences (as shown in the above example). In contrast, our Sim-DS is a particular type of DS method where X and Y are the same types of data—in such a case, we can infer labels for Y based on X's label and the similarity between X and Y. Without loss of generality, we define Sim-DS as:

> **Sim-DS**: given X with the label L (from a KB), assigns L to Y (a data entry in the target corpus) if X and Y are similar to each other.

Particularly, our Sim-DS method for definition sentence retrieval is a Sim-DS problem where: 1) both X and Y are sentences; 2) X is a positive instance (in our case, a definition sentence).

> **Sim-DS for annotating definition sentences**: let X (from a KB) and Y (from a target corpus) be sentences with the occurrence of the same target term t; given that X is a definition sentence of t,

we annotate the quality of Y for explaining t by $sim(X, Y)$.

Figure 1 (right) shows an example of the Sim-DS method used for annotating definition sentences. First, we extract X, a definition sentence for the target term *Power Electronics*, from Wikipedia (our choice of KB in this example). Then, we annotate two sentences Y_1 and Y_2 from the target corpus based on their text similarities with X. In this example, Sim-DS assigns Y_2 a higher quality score than Y_1 because Y_2 shares more linguistic units with X than Y_1 does (a higher level of text similarity).

Our Sim-DS problem is challenging and interesting in that:

- We need an appropriate text similarity measure that works well for *the annotation task*. Although text similarity techniques have been widely studied in many scenarios, we know little about whether they can help us assess the quality of definition sentences.
- In many practical situations, we can only rely on heuristics to obtain *imperfect* example definition sentences (X) from a KB, which makes Sim-DS even more challenging.
- Sim-DS offers a chance to *combine automatic annotation with manual judgments*. As later sections introduced, we can use manual judgments to train refined textual similarity measures to improve the quality of Sim-DS annotation, which is more effective than directly using the same set of judgments to train definition sentence retrieval methods.

4.2 Knowledge Base and Example Extraction

The first step of Sim-DS is to determine the knowledge base (KB) used for annotation. This needs to take into account two issues in the case of our problem: 1) we need to be able to reliably extract high-quality example definition sentences from the KB; 2) the KB needs to have a sufficient overlap with the target corpus in topic, such that we can generate a sufficient amount of training data.

We use Wikipedia as the KB to generate judgments for definition sentences in this study. We also believe Wikipedia is an effective choice of KB for generating judgments of definition sentences in many corpora because 1) the first sentence of a Wikipedia entry is usually a high-quality definition sentence, and 2) Wikipedia covers a broad range of topics. We do not further discuss other choices of KB because this is highly problem-dependent.

We manually judged 100 randomly selected Wikipedia entries' first sentences using the criteria in Table 1. Among these sentences, 67 were judged as "3", 14 as "2", 14 as "1", and 5 as "0". 95% of the sentences satisfy the minimum requirement of definition sentences. Ideal definition sentences ("3") take up 67% of the judged sentences. However, we also note that about 1/3 of these sentences are imperfect ones (not informative enough or not definitional).

After we extracted example definition sentences from the KB, we can further retrieve candidate sentences from the target corpus using the target terms and apply text similarity measures to annotate these sentences such that we can train ranking models.

4.3 Unsupervised Text Similarity Measures

A straightforward choice of implementing Sim-DS is to adopt an unsupervised text similarity measure for annotation. We call this approach **unsupervised Sim-DS**. Unsupervised Sim-DS can generate training instances without any manual judgments. We examine several representative unsupervised text similarity measures. Note that we exclude the target term when computing these measures.

BOW-cosine. The first similarity measure we examined is the cosine similarity of two sentences based on their bag-of-words representations (with TF-IDF weighting). We remove stop words (the standard stop word list in Lucene) and apply Krovertz stemming [18] when computing BOW cosine similarity.

ROUGE-SU9. ROUGE [20] is a family of text similarity measures initially proposed to evaluate automatic summarization systems. It evaluates machine-generated summaries by comparing them to human-edited ones and prefers those that are similar to the ground truth ones.

We use an IDF weighted ROUGE measure as in Equation 1, where: u refers to a linguistic unit, such as a unigram, a bigram, a skip gram, and so on; Y is a sentence from the target corpus, which is to be annotated by the Sim-DS; X is an example sentence from Wikipedia (a reference sentence); $c(u, X)$ and $c(u, Y)$ is the frequency of u in X and Y; IDF(u) is computed as the sum of IDF for each word in u (for example, if u is a bigram, IDF(u) is the sum of the IDF for the two words in the bigram).

$$\text{ROUGE}_{\text{precision}} = \frac{\sum_u \min(c(u, Y), c(u, X)) \cdot \text{IDF}(u)}{\sum_u c(u, Y) \cdot \text{IDF}(u)} \quad (1)$$

$$\text{ROUGE}_{\text{recall}} = \frac{\sum_u \min(c(u, Y), c(u, X)) \cdot \text{IDF}(u)}{\sum_u c(u, X) \cdot \text{IDF}(u)}$$

We use a version of ROUGE called ROUGE-SU9 for Sim-DS, where u is a combination of unigrams and skip grams within a distance of 9 words. We made a few modifications to the original measures [20] to better cope with our Sim-DS problem:

- We only consider nouns and adjectives. This is to reduce annotation bias. For example, if the Wikipedia reference sentence uses the pattern "`is defined as`", it will inflate scores of target sentences using the same pattern. Only considering nouns and adjectives reduces such bias.
- Many previous studies of text summarization used ROUGE recall for evaluation, while we use ROUGE F-measure with an equal weight on ROUGE precision and recall ($\beta = 1$). As Section 6.2 will discuss, this reduces the bias of Sim-DS annotation about the length of the target sentences.

CBOW. We also apply word embeddings to Sim-DS annotation, which may help address the vocabulary mismatch problem when measuring textual similarities. Let X and Y be two sentences. We represent X and Y as the sum of word vectors for each word (weighted by the IDF of the words) and compute the cosine similarity of the two sentence vectors. We use a pre-trained CBOW model (300 dimensions) based on Google News data[3].

ROUGE-CBOW-SU9. Ng et al. [28] developed a variant of ROUGE, which allows semantic matching based on word embeddings. In contrast to regular ROUGE measures, it allows matching of different linguistic units (u) based on word embeddings. Here our ROUGE-CBOW-SU9 measure uses a pre-trained CBOW model. We also

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

retain the other settings as the same as we applied to ROUGE-SU9, e.g., IDF weighting, only considering nouns and adjectives, and computing ROUGE F-score.

We also note that the four options of unsupervised text similarity measures also cover different types of textual similarity techniques. BOW-cosine does not consider word dependency or semantic matching. ROUGE-SU9 considers word dependency because it uses skip grams. CBOW and ROUGE-CBOW-SU9 consider semantic matching using state-of-the-art distributed word representation. This also allows us to examine the contribution of word dependency and semantic matching in Sim-DS annotation.

	Word dependency	Semantic Matching
BOW-cosine		
ROUGE-SU9	✓	
CBOW		✓
ROUGE-CBOW-SU9	✓	✓

4.4 Supervised Text Similarity Measures

While manual judgments are available, we can train supervised text similarity measures for Sim-DS annotation. We call this approach **supervised Sim-DS**. Supervised Sim-DS stands for a novel way of combining automatic annotation and manual judgments. Our supervised text similarity measures are based on regression models using 31 features. These features include the unsupervised similarity measures introduced in Section 4.3 as well as several well-performing methods [4, 36] reported in the 2016 SemEval Semantic Textual Similarity (STS) evaluation [1].

Table 2 lists the 31 features, including 15 variants of ROUGE, 2 variants of the word alignment model by Sultan et al. [36], 6 features comparing the two sentences by their part-of-speech (POS) tags, 4 variants of bag-of-words cosine similarity, and 5 semantic matching features using different distributed representations of words [23, 31] and sentences [19]. Similar to unsupervised measures, we exclude the target term when computing these features.

We examine supervised text similarity measures learned from two different types of training data as follows:

Using External Training Data. We train supervised text similarity measures using the data of the SemEval STS task from 2012 to 2016, including 6,683 judged sentence pairs (we only use the English language training data). Note that the SemEval STS task [1] aims at developing methods to determine the closeness of two sentences in meaning, which is different from our task. The training and testing data consist of pairs of sentences, along with manually judged scores (ranging from 0 to 5) indicating the closeness of the two sentences in meaning.

Note that the form of training data in the SemEval STS task is slightly different from the requirement of our task—we exclude the target term when computing text similarity measures. To fit the SemEval STS data into our problem, we match the common noun phrase between each pair of sentences in the STS dataset and treat the matched noun phrase as the target term, such that we can train text similarity measures. As we will discuss in Section 6 and Section 7, despite the different nature of STS and our purpose, we can learn effective Sim-DS techniques using this type of external training data.

Table 2: Features for supervised textual similarity measures.

Feature Group	#	Description
Word-overlap	15	Variants of ROUGE-N1, ROUGE-N2, ROUGE-L, and ROUGE-SU9 using ROUGE precision, recall, or F1, considering all words or only nouns and adjectives.
Word-alignment	2	Weighted and unweighted version of Sultan et al.'s word alignment model [36], which aligns the common part of two sentences and compute its proportion.
POS-overlap	6	The proportion of overlapping part-of-speech (POS) tags between two sentences, considering unigrams, bigrams, and trigrams.
BOW-cosine	4	Variants of bag-of-words cosine, w/ or w/o TF-IDF weighting, and w/ or w/o stemming.
Semantic	5	Cosine similarity of sentence vectors based on pretrained CBOW [23], SkipGram [23], Glove [31], and Paragraph2Vec [19] models.

Using Local Training Data. We sampled 500 sentences from our target corpus and manually judged their quality as definition sentences (Section 6.1 introduces the details). We train supervised text similarity measures on this small set of sentences. This small set of manual judgments is not enough to train effective definition sentence retrieval models directly, but can significantly improve Sim-DS (Section 7 reports the results).

We train linear regression models (with L2-Regularization) and ν-SVR models (with RBF kernel) using the two types of training data. We compare them with the unsupervised methods in Section 6 and Section 7.

Note that the described supervised similarity measures are representative of the state-of-the-art text similarity techniques according to our experiments on the SemEval STS 2016 dataset [1]. Using the SemEval STS 2012–2015 dataset for training, our method can achieve 0.749 prediction correlation (the correlation between the predicted scores and the judged ones) on the SemEval 2016 test set. The performance could be ranked at the 6th place (out of 113 runs) in the SemEval STS 2016 evaluation [1].

5 DATASET

We evaluate Sim-DS on a corpus of computer science articles. The corpus includes 277,933 articles from the ACM digital library. We followed the list of ACM conference proceedings[4] and accessed the PDF documents of articles in 2015. We extracted full texts from the PDF documents. We annotated part-of-speech (POS) tags using the Stanford NLP toolkit [22] and chunked texts using OpenNLP[5].

We use a Wikipedia corpus as the knowledge base (KB) in our study. The corpus includes a dump of the English-language Wikipedia in March 2017. To produce Sim-DS judgments, We matched Wikipedia entries with noun phrases in the ACM corpus. For each term that is both a Wikipedia entry and a noun phrase in the ACM corpus, we use the Wikipedia entry's first sentence to generate Sim-DS judgments for the sentences in the ACM corpus with the occurrence of that term. We excluded some Wikipedia entries and noun phrases to ensure the quality of Sim-DS judgments:

[4] http://dl.acm.org/proceedings.cfm
[5] http://opennlp.apache.org/

- We only consider Wikipedia articles that include the word "computer", "information", or "data". This selects a subset of Wikipedia entries close to the topic of the ACM corpus.
- We exclude ambiguous Wikipedia entries (entries with a disambiguation page in Wikipedia).
- If the Wikipedia entry's title includes only one word, we exclude it if the word is too common (by its frequency in the Google 5-gram dataset).
- We excluded noun phrases without nouns extracted from the ACM corpus (e.g., "we" and "I").
- We excluded noun phrases that are too rare (appeared in fewer than 10 articles) or too common in the target ACM corpus (with IDF < 5).

We evaluate our Sim-DS methods from two aspects:

- by the accuracy of the automatically generated judgments (Section 6);
- by the effectiveness of the definition retrieval models trained using the Sim-DS judgments (Section 7).

6 EVALUATION I: ACCURACY AND BIAS

6.1 Evaluation Method

We selected 500 sentences from the ACM corpus and manually judged them regarding their quality as definition sentences. The 500 sentences were chosen using the following procedure. We randomly selected 25 target terms among the overlapping ones between the ACM corpus and the Wikipedia collection. For each target term, we further sampled 20 sentences with the occurrence of the term from the ACM corpus. Among the 20 sentences, ten were randomly selected from the top 5 scored sentences of each Sim-DS methods we introduced in Section 4. The other ten sentences were randomly sampled from the rest. This was to ensure that the selected sentences have both definitional and non-definitional ones.

We manually judged the 500 sentences using the criteria in Table 1. We evaluate Sim-DS methods by the following criteria:

- **Annotation accuracy** – evaluated by the correlation of the Sim-DS judgments compared with human assessments. We use Spearman's ρ (a rank correlation measure) because we are mainly interested in to which extent the ranking of sentences by the Sim-DS judgments agrees with that by human assessments. We separately compute the correlation values for the 20 sentences of each target term. Then, we report the average correlation of the 25 terms.
- **Length bias** – the correlation (Spearman's ρ) of the Sim-DS judgments with sentence length. Some similarity measures may produce biased judgments such as preferring long or short sentences, which is not ideal for training definition sentence retrieval models.

6.2 Unsupervised Measures

Table 3 reports the annotation accuracy and length bias of Sim-DS methods using the unsupervised textual similarity measures introduced in Section 4.3.

Among the examined measures, ROUGE-CBOW-SU9 produced the most accurate judgments ($\rho = 0.551$), which is also not surprising considering that ROUGE-CBOW-SU9 takes into account both term dependency and semantic matching. ROUGE-CBOW-SU9 sightly outperformed ROUGE-SU9 in terms of the correlation with human judgments ($\rho = 0.551$ vs. $\rho = 0.541$), suggesting that word embeddings are useful to regular ROUGE measures for the purpose of judging definition sentences. ROUGE-SU9 also outperformed BOW-cosine ($\rho = 0.541$ vs. $\rho = 0.489$) and ROUGE-N1 ($\rho = 0.502$), suggesting that the combination of unigram and skip gram features in ROUGE-SU9 is helpful to Sim-DS methods.

Regarding length bias, we note that CBOW tends to assign higher scores to longer sentences—the Sim-DS judgments have a moderate correlation $\rho = 0.407$ with sentence length. In contrast, the other three unsupervised measures did not show significant bias towards long or short sentences. However, combining CBOW with ROUGE-SU9 does not show such length bias.

Different ROUGE Variants. We further note that our modifications to the original ROUGE measures are necessary and important.

Table 4 compares Sim-DS methods using different variants of ROUGE-SU9, where "Only NN JJ" refers to ROUGE measures using only nouns and adjectives. First, we found that only considering nouns and adjectives is very helpful, enhancing the correlation between Sim-DS judgments and human assessments by 0.22 (from $\rho = 0.319$ to $\rho = 0.541$). Second, we note that ROUGE precision and recall have significant length bias, while ROUGE F1 sets off such bias—ROUGE precision prefers longer sentences ($\rho = 0.197$ with

Table 3: Annotation accuracy and bias of Sim-DS methods using different unsupervised textual similarity measures.

Unsupervised Similarity Measures	Correlation w/ human judgments (annotation accuracy)	Correlation w/ sentence length (length bias)
BOW-cosine	0.489	−0.013
CBOW	0.399	**0.407**
ROUGE-SU9	0.541	−0.017
ROUGE-CBOW-SU9	**0.551**	−0.035

Table 4: Annotation accuracy and bias of Sim-DS methods using different variants of ROUGE-SU9.

ROUGE-SU9 Variants	Correlation w/ human judgments (annotation accuracy)	Correlation w/ sentence length (length bias)
Only NN JJ, F1	**0.541**	−0.017
All words, F1	0.319	−0.088
Only NN JJ, precision	0.528	0.197
Only NN JJ, recall	0.507	**−0.209**

Table 5: Annotation accuracy and bias of Sim-DS methods using similarity measures with different word embeddings.

Similarity Measures	Correlation w/ human judgments (annotation accuracy)	Correlation w/ sentence length (length bias)
CBOW	0.399	0.407
SkipGram	**0.404**	**0.443**
Glove	0.361	0.331
ROUGE-CBOW-SU9	**0.551**	−0.035
ROUGE-SkipGram-SU9	0.550	**−0.051**
ROUGE-Glove-SU9	0.528	−0.024

Table 6: Correlation (Spearman's ρ) of Sim-DS sentence scores with human judgments and sentence lengths.

		External (SemEval STS)		Local (ACM corpus)		SemEval STS 2016 results
		Correlation w/ human judgments (annotation accuracy)	Correlation w/ sentence length (length bias)	Correlation w/ human judgments (annotation accuracy)	Correlation w/ sentence length (length bias)	
ν-SVR	All features	**0.594**	0.091	**0.606**	0.045	**0.749**
Lin-Reg	All features	0.579	0.052	0.599	0.049	0.741
	Word-overlap	**0.565**	−0.101	0.561	−0.092	0.723
	Word-alignment	0.550	0.072	**0.564**	0.052	**0.734**
ν-SVR	POS-overlap	−0.037	0.026	0.172	0.031	0.303
	BOW-cosine	0.480	−0.013	0.505	0.042	0.682
	Semantic	0.391	**0.390**	0.352	**0.403**	0.669

sentence length), while ROUGE recall tends to assign higher scores to shorter sentences ($\rho = -0.209$ with sentence length).

Choice of Word Embeddings. Table 5 further compares Sim-DS using different word embeddings, where SkipGram and Glove refer to the cosine similarity of sentence vectors based on the sum of individual words' SkipGram and Glove vectors. We found that different word embeddings do not differ much regarding Sim-DS annotation accuracy. However, the cosine similarity of sentence vectors consistently showed a significant length bias regardless of which word embedding models were employed. Nevertheless, ROUGE measures using these word embeddings did not show such length bias.

To sum up, results in this section show that unsupervised textual similarity measures such as ROUGE-SU9 and ROUGE-CBOW-SU9 (using F1 scores and only considering nouns and adjectives) can produce accurate Sim-DS judgments that have a moderate correlation with human judged definition sentence quality. In addition, some similarity measures have significant risks of favoring long or short sentences, which should be avoided in Sim-DS methods.

6.3 Supervised Measures

Table 6 reports the annotation accuracy and bias of supervised Sim-DS methods. We also report the performance of the similarity measures in the SemEval STS 2016 task as a reference (the last column), where the reported numbers are the Pearson's correlation between predicted and judged semantic similarity scores (the official evaluation measure of the SemEval STS 2016). Results show that supervised text similarity measures, regardless of training using external or local data, can effectively help Sim-DS generate more accurate judgments than the unsupervised ones.

As Table 6 shows, both the ν-SVR and the linear regression models (using all features) outperform the unsupervised ones. Compared with the unsupervised measures, the supervised ones have about 0.05 higher Spearman's correlation with human judgments. The supervised measures also did not show significant length bias. The ν-SVR model slightly outperforms the linear regression model. Thus, we use ν-SVR in following experiments.

According to Table 6, supervised Sim-DS methods trained using external and local data are comparable regarding annotation accuracy. This suggests that, although the purpose of the SemEval STS task is very different from that of our problem, it is still helpful to our Sim-DS method. This is an important finding because it suggests that we do not necessarily need to collect training data for the particular Sim-DS tasks to apply supervised DS—instead, we can effectively improve the performance of unsupervised Sim-DS based on generic-purpose text similarity training data such as those from the SemEval STS task.

Also, we note that the performance of different supervised similarity measures in the SemEval STS task is consistent with the annotation accuracy of Sim-DS methods using these measures. This further suggests that generic-purpose text similarity benchmark such as SemEval STS can provide consistent help to Sim-DS annotation. Our Sim-DS method may further benefit from better text similarity techniques tested on benchmarks such as SemEval STS.

7 EVALUATION II: DEFINITION RETRIEVAL

7.1 Evaluation Setting

This section evaluates Sim-DS methods by the effectiveness of definition sentence retrieval models trained using the Sim-DS judgments. We train two representative definition sentence retrieval models:

- **Bigram** [8]: using a target term's surrounding bigrams to determine whether or not the sentence explains that term.
- **Learning-to-rank**: a LambdaMART [5] learning-to-rank model with 248 features, including 216 variants of the bigram models' scores (by considering bigrams with different distances to the target term, etc.) and 32 other features (such as the length of the sentence and the proportion of stop words in the sentence). To avoid over-fitting, we use 1/4 of the judgments to train bigram models to compute the features and the rest 3/4 to train ranking model.

We compare Sim-DS judgments with manual and heuristic judgments. We train the bigram and learning-to-rank models using the following types of training data:

- **Manual, external**: manual definition sentence judgments from a corpus other than the target ACM corpus. We use two public datasets: Jin [16] includes 2,184 sentences from the ACL anthology corpus; WCL [26] includes 4,719 judged sentences from Wikipedia.
- **Manual, local**: manual definition sentence judgments on the target ACM corpus. We examine two sets of judgments: ACM1 includes the 500 judged sentences used in Section 6, which is also the set of judgments we used for training supervised Sim-DS; ACM2 contains 1,732 sentences from the top 3 results of different runs evaluated in this section.
- **Heuristic**: heuristically judged definition sentences from Wikipedia. We consider the first sentence of a Wikipedia

Table 7: 12 annotators' evaluation of definition sentence quality for 50 terms.

Ranking Method	Training Data	Type	Size	Average Rating		Precision		nDCG@3
				top 1	top 3	@1	@3	
DefMiner [16]	Jin (ACL anthology)	manual, external	2.1K	0.82	0.56	0.48	0.33	0.377
WCL-3 [26]	WCL (Wikipedia)	manual, external	4.7K	0.66	0.60	0.42	0.36	0.350
Bigram [8]	[1] Jin (ACL anthology)	manual, external	2.1K	0.58	0.47	0.34	0.28	0.300
	[2] WCL (Wikipedia)	manual, external	4.7K	0.64	0.51	0.38	0.31	0.313
	[3] ACM1	manual, local	500	0.46	0.37	0.24	0.21	0.250
	[4] ACM2	manual, local	1.7K	0.59	0.48	0.34	0.29	0.302
	[5] Wiki, first sentence	heuristic	117K	0.60	0.59	0.36	0.37	0.343
	[6] ROUGE-CBOW-SU9	sim-DS, unsup	1.8M	0.74^{12345}	0.62^{134}	0.44^{134}	0.39^{1234}	0.371^{12345}
	[7] ν-SVR, external	sim-DS, sup	1.8M	$\mathbf{0.84}^{12345}$	0.67^{1234}	$\mathbf{0.50}^{12345}$	0.41^{1234}	0.388^{12345}
	[8] ν-SVR, local	sim-DS, sup	1.8M	0.82^{12345}	$\mathbf{0.69}^{1234}$	0.46^{12345}	$\mathbf{0.41}^{1234}$	$\mathbf{0.396}^{12345}$
Learning-to-rank	[3] ACM1	manual, local	500	0.11	0.09	0.08	0.07	0.091
	[4] ACM2	manual, local	1.7K	0.12	0.10	0.10	0.09	0.125
	[5] Wiki, first sentence	heuristic	117K	0.86	0.59	0.52	0.36	0.375
	[6] ROUGE-CBOW-SU9	sim-DS, unsup	1.8M	1.00^{345}	0.73^{345}	0.64^{345}	0.46^{345}	0.446^{345}
	[7] ν-SVR, external	sim-DS, sup	1.8M	$\mathbf{1.10}^{345}$	0.77^{345}	$\mathbf{0.66}^{345}$	$\mathbf{0.49}^{345}$	0.477^{345}
	[8] ν-SVR, local	sim-DS, sup	1.8M	1.08^{345}	$\mathbf{0.78}^{345}$	0.62^{34}	0.47^{345}	$\mathbf{0.487}^{345}$

^{12345678} indicate the result is significantly different from the numbered runs at 0.05 level by paired t-test.

article as a definition sentence. Further, we count the rest of the sentences with the occurrence of the Wikipedia entry as non-definitional (based on the assumption that a term does not need to be defined twice in an article). This corpus includes 117,553 sentences extracted from the 5,537 Wikipedia entries matched with the ACM corpus (as we described in Section 5). We also use the positive instances of this dataset as references to produce Sim-DS judgments.

- **Sim-DS**: sentences from the ACM corpus that are automatically judged using Sim-DS methods. Each Sim-DS dataset includes 1.8 million automatically judged sentences.

We also report results from two pre-trained definition sentence retrieval models as a reference: the word-class lattice (WCL-3) model [26] trained using the WCL dataset, and DefMiner [16] trained using the Jin dataset. These two models need to be trained using word-level annotations (such as whether or not a word belongs to a term or its definition). They cannot be trained using the sentence-level Sim-DS judgments.

Table 8: Examples of terms used for evaluation.

passive RFID, backscatter communication, consensus routing, OSPF, BlinkDB, approximate query processing, data provenance, multi-query optimization, privacy budget, aperture problem, ImageNet, attention model, dependency parsing, BM25F

12 Computer Science Ph.D. students participated in the evaluation of the definition sentence retrieval models. We first asked each participant to provide 3 to 5 terms related to his/her research domain. We specifically requested that they provide terms that do not have a Wikipedia page to evaluate how well our approach performs on less common terms that do not exist in Wikipedia (since we used Wikipedia sentences as references to produce Sim-DS judgments). We collected 50 terms from the participants in total. We find and rank definition sentences for the each terms using each model. We generate a judgment pool including the top 3 ranked sentences by each model. The participants assessed the quality of the sentences for the terms they proposed (we presented the sentences to them in random order) using the criteria in Table 1. Table 8 shows some examples of the 50 target terms.

7.2 Retrieval Effectiveness

Table 7 reports the retrieval effectiveness of Bigram and Learning-to-rank models trained using different types of judgments based on participants' evaluation of top-ranked sentences. We report: the average rating of sentences at the top 1 and top 3 ranks; the precision of the sentence being definitional (rating > 0) at the top 1 and top 3 ranks; normalized discounted cumulative gain (nDCG) [15] of the top 3 sentences. All measures agree that models trained using the Sim-DS judgments perform better than others.

Sim-DS vs. Manual. Models trained using Sim-DS judgments of 1.8 million sentences on the target corpus performed consistently better than those trained using manual judgments of only a few thousand sentences, regardless of whether the manual judgments were collected on the target corpus (ACM1 and ACM2) or not (Jin and WCL). Note that Jin and WCL are representative of the typical amount of annotation efforts in the research community. This suggests that collecting manual training instances is practically limited in its scale, which is usually difficult to train optimal models. In contrast, Sim-DS can quickly produce large-scale training instances. Although the judgments are not perfectly accurate, the scale of Sim-DS judgments is usually large enough to train more effective definition sentence retrieval models than manual judgments. Note that neither the unsupervised Sim-DS nor the supervised one using external training data (SemEval STS) requires any manual judgments on the target corpus. But both methods trained more effective definition sentence retrieval models than using manual judgments. We believe it is because Wikipedia provides large-scale

and high-quality knowledge for solving our task. With appropriately designed methods, such knowledge is even more useful than explicitly labeled data for solving the task.

Sim-DS vs. Heuristics. Models trained using noisy, yet large and localized Sim-DS judgments perform consistently better than those trained using more accurate, smaller (but still much large compared with manual annotations), and non-localized heuristic judgments on the Wikipedia corpus (Wiki). Note that in our Sim-DS method, we generated the Sim-DS judgments using the positive instances of Wikipedia judgments based on text similarity measures. Considering that the text similarity measure used for the Sim-DS annotation is not perfect, we believe the generated Sim-DS judgments should be less accurate than the Wikipedia judgments. Whereas the Sim-DS judgments indeed trained more effective definition sentence retrieval models. This indicates that: 1) Sim-DS serves as an effective "bridge", transforming annotations from an external corpus to large and more useful localized annotations on the target corpus; 2) even noisy and imperfect localized annotations seem more useful than more accurate, but non-localized labels. We discuss more details in the next section.

Sim-DS Variants. Results suggest that supervised Sim-DS can consistently generate more useful definition sentence judgments than the unsupervised ones. Models trained using the supervised Sim-DS judgments consistently outperform models trained using the unsupervised ones, although the difference is not statistically significant (probably due to the limited size of our test collection).

Second, we found that supervised Sim-DS trained using external (SemEval STS) and local training data (ACM1) generated comparably useful definition sentence judgments—models trained using two types of judgments perform very close to each other. This is also consistent with the findings in Section 6. This further suggests that general-purpose textual similarity techniques (such as those studied in the SemEval STS tasks) are useful for improving Sim-DS methods. However, we note that the external training data (SemEval STS) are much larger than the local ones (ACM 1 includes only 500 sentences). Thus, we believe our experiments are not conclusive enough to determine.

To conclude, results in this section further shows that Sim-DS methods do generate useful judgments that are helpful for training definition sentence retrieval models. The unsupervised Sim-DS offers a cheap solution for definition retrieval

7.3 Analysis of Definition Sentence Patterns

In addition to its large scale, Sim-DS also has a key advantage—it produces *localized* training instances on the target corpus, which helps learn representative models pertinent to the target corpus.

To illustrate this advantage of Sim-DS, we compare the bigram models learned from different judgments with those learned from ACM2—manual judgments on the target ACM corpus. Despite that ACM2 is relatively small in size, we believe its top-ranked bigrams should still be representative of definition sentences in the target ACM corpus. We compare two bigram models by the rank correlation (Spearman's ρ) of their top 500 bigrams. Note that we use rank correlation to compare two bigram models because it is intuitive to interpret correlation strength. We examined and found that we can come to similar conclusions using other methods such as KL-divergence for comparison.

Table 9: Rank correlation (Spearman's ρ) of top 500 bigrams learned from different datasets.

	ACM2	Sim-DS, sup	Sim-DS, unsup	Wiki	Jin (ACL)
Sim-DS, sup	0.43	-	-	-	-
Sim-DS, unsup	0.35	0.76	-	-	-
Wiki	−0.02	0.16	0.18	-	-
Jin (ACL)	−0.02	0.29	0.27	0.12	-
WCL	−0.07	0.12	0.09	0.41	0.06

Table 10: The ranks of bigram patterns in models learned from different sets of judgments.

Bigram Patterns	Rank of patterns (by probability) in …			
	Sim-DS, sup	ACM2	Wiki	Jin (ACL)
`* is a`	1	1	1	1
`* is the`	2	3	2	2
`such as *`	3	33	1802	396
`* , a`	4	10	46	11
`* is an`	5	2	3	10
`* [ 2`	26	21	-	-
`* [ 5`	27	66	-	4807
`* [ 3`	32	36	-	-
`science , *`	138	215	6	1783
`computing , *`	603	-	14	-

Table 9 reports the rank correlation of bigram models learned from different judgments. We found that only those learned from Sim-DS judgments ("Sim-DS, sup" and "Sim-DS, unsup") have a positive correlation ($\rho = 0.43$ and $\rho = 0.35$) with the model learned from ACM2. ACM2's bigram model does not agree much with those learned from Wiki, Jin, or WCL. This suggests that it is risky to generalize models learned from one corpus to another. In contrast, Sim-DS directly generate judgments on the target corpus. Such localized judgments (even if they are not perfectly accurate) help train models that are representative of the target corpus.

Table 10 further shows some examples by listing the ranks of bigrams in models learned from different judgments. We use `*` for the target term, and we only look into bigrams right before or after the target term. It shows that different models do not agree with each other on the importance of the bigram patterns (except for a few popular patterns such as "`* is a`").

The bigram model learned from the Wiki dataset is greatly influenced by the style of Wikipedia. For example, "`science , *`" and "`computing , *`" are popular bigrams in the Wiki dataset because many Wikipedia articles included sentences such as "*In computer science, * is a …*". Unsurprisingly, these patterns do not guarantee to generalize to other corpora. In contrast, Sim-DS helps learn bigram patterns that are representative of the target ACM corpus. For example, we found that bigrams such as "`* [ 2`", "`* [ 5`", and "`* [ 3`" are highly ranked in ACM2. This is because when defining a term, authors usually cited to the initial article that had proposed that term (the ACM proceeding templates use a numbered citation format), e.g., "*distant supervision [2] refers to …*". Such patterns do not exist in the Wiki corpus. They are also very rare in the Jin dataset (based on the ACL anthology corpus) because ACL articles

used a different citation format. However, these patterns are also highly ranked in the Sim-DS judgments.

These examples further demonstrate that the Sim-DS approach can generate localized annotations specific to the target corpus, which is a significant advantage making Sim-DS a technique that is capable of generalizing to corpora of different domains.

8 CONCLUSION

Two practical concerns for supervised approaches are the high cost of collecting human judgments and the limited generalizability to new corpora. We proposed a similarity-based distant supervision method to address these issues for the task of definition sentence retrieval. Experimental results verified a few advantages of Sim-DS:

- *Low-cost* – Our method performs well without any human judgments. It can be further improved significantly with the help of either existing training data for semantic textual similarity tasks or only a few hundred judged sentences for our task (which are not sufficient to train supervised solutions directly).
- *Effectiveness* – Both unsupervised and supervised Sim-DS train highly accurate definition sentence retrieval models that outperform those using manual or heuristic judgments.
- *Flexibility* – Sim-DS always generates judgments on the target corpus. The broad coverage of Wikipedia also makes it easy to apply our method to corpora of different domains.

Admittedly, our work also has a few limitations. For example, we did not experiment on other domains to adequately demonstrate the generalizability of our method. Another limitation is that we stay at generating sentence-level judgments, which does not directly help a closely related task—definition extraction. Also, we did not compare our method to other weakly supervised methods for this task [10]. We leave these issues for future work.

Resources related to this study can be accessed online[6].

ACKNOWLEDGMENT

This work was supported in part by the Center for Intelligent Information Retrieval and in part by NSF grant #IIS-0910884. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

REFERENCES

[1] E. Agirre, C. Banea, D. Cer, M. Diab, A. Gonzalez-Agirre, R. Mihalcea, G. Rigau, and J. Wiebe. SemEval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *Proceedings of SemEval-2016*, pages 497–511, 2016.

[2] M. S. Bernstein, J. Teevan, S. Dumais, D. Liebling, and E. Horvitz. Direct answers for search queries in the long tail. In *CHI '12*, pages 237–246, 2012.

[3] G. Boella and L. Di Caro. Extracting definitions and hypernym relations relying on syntactic dependencies and support vector machines. In *ACL '13*, pages 532–537, 2013.

[4] T. Brychcın and L. Svoboda. UWB at SemEval-2016 task 1: Semantic textual similarity using lexical, syntactic, and semantic information. In *Proceedings of SemEval-2016*, pages 588–594, 2016.

[5] C. J. Burges. From RankNet to LambdaRank to LambdaMART: An overview. Technical Report MSR-TR-2010-82, Microsoft Research, 2010.

[6] B. Carterette and J. Allan. Semiautomatic evaluation of retrieval systems using document similarities. In *CIKM '07*, pages 873–876, 2007.

[7] L. B. Chilton and J. Teevan. Addressing people's information needs directly in a web search result page. In *WWW '11*, pages 27–36, 2011.

[8] H. Cui, M.-Y. Kan, and T.-S. Chua. Generic soft pattern models for definitional question answering. In *SIGIR '05*, pages 384–391, 2005.

[9] H. Cui, M.-Y. Kan, and T.-S. Chua. Soft pattern matching models for definitional question answering. *ACM Transactions on Information Systems*, 25(2), 2007.

[10] L. Espinosa-Anke, F. Ronzano, and H. Saggion. Weakly supervised definition extraction. In *Proceedings of Recent Advances in Natural Language Processing*, pages 176–185, 2015.

[11] L. Espinosa-Anke and H. Saggion. Applying dependency relations to definition extraction. In *Proceedings of the 19th International Conference on Applications of Natural Language to Information Systems*, pages 63–74, 2014.

[12] M. A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *COLING '92*, pages 539–545, 1992.

[13] E. Hovy, C.-Y. Lin, L. Zhou, and J. Fukumoto. Automated summarization evaluation with basic elements. In *LREC '06*, pages 899–902, 2006.

[14] A. Intxaurrondo, E. Agirre, O. L. de Lacalle, and M. Surdeanu. Diamonds in the rough: Event extraction from imperfect microblog data. In *NAACL '15*, pages 641–650, 2015.

[15] K. Järvelin and J. Kekäläinen. IR evaluation methods for retrieving highly relevant documents. In *SIGIR '00*, pages 41–48, 2000.

[16] Y. Jin, M.-Y. Kan, J.-P. Ng, and X. He. Mining scientific terms and their definitions: A study of the ACL anthology. In *EMNLP '13*, pages 780–790, 2013.

[17] K.-W. Kor and T.-S. Chua. Interesting nuggets and their impact on definitional question answering. In *SIGIR '07*, pages 335–342, 2007.

[18] R. Krovetz. Viewing morphology as an inference process. In *SIGIR '93*, pages 191–202, 1993.

[19] Q. Le and T. Mikolov. Distributed representations of sentences and documents. In *ICML '14*, pages 1188–1196, 2014.

[20] C.-Y. Lin. ROUGE: A package for automatic evaluation of summaries. In *ACL '04 Workshop on Text Summarization Branches Out*, 2004.

[21] C.-Y. Lin and E. Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *NAACL '03*, pages 71–78, 2003.

[22] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky. The Stanford CoreNLP natural language processing toolkit. In *ACL '14*, pages 55–60, 2014.

[23] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS '13*, pages 3111–3119, 2013.

[24] M. Mintz, S. Bills, R. Snow, and D. Jurafsky. Distant supervision for relation extraction without labeled data. In *ACL '09*, pages 1003–1011, 2009.

[25] S. Muresan and J. Klavans. A method for automatically building and evaluating dictionary resources. In *LREC '02*, pages 231–234, 2002.

[26] R. Navigli and P. Velardi. Learning word-class lattices for definition and hypernym extraction. In *ACL '10*, pages 1318–1327, 2010.

[27] A. Nenkova and R. Passonneau. Evaluating content selection in summarization: The pyramid method. In *NAACL '04*, 2004.

[28] J.-P. Ng and V. Abrecht. Better summarization evaluation with word embeddings for ROUGE. In *EMNLP '15*, pages 1925–1930, 2015.

[29] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. BLEU: A method for automatic evaluation of machine translation. In *ACL '02*, pages 311–318, 2002.

[30] R. J. Passonneau, E. Chen, W. Guo, and D. Perin. Automated pyramid scoring of summaries using distributional semantics. In *ACL '13*, pages 143–147, 2013.

[31] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *EMNLP '14*, pages 1532–1543, 2014.

[32] M. Purver and S. Battersby. Experimenting with distant supervision for emotion classification. In *EACL '12*, pages 482–491, 2012.

[33] M. Reiplinger, U. Schäfer, and M. Wolska. Extracting glossary sentences from scholarly articles: A comparative evaluation of pattern bootstrapping and deep analysis. In *Proceedings of the ACL-2012 Special Workshop on Rediscovering 50 Years of Discoveries*, pages 55–65, 2012.

[34] X. Ren, A. El-Kishky, C. Wang, F. Tao, C. R. Voss, and J. Han. ClusType: Effective entity recognition and typing by relation phrase-based clustering. In *KDD '15*, pages 995–1004, 2015.

[35] R. Snow, D. Jurafsky, and A. Y. Ng. Learning syntactic patterns for automatic hypernym discovery. In *NIPS '04*, pages 1297–1304, 2004.

[36] M. A. Sultan, S. Bethard, and T. Sumner. DLS@CU: Sentence similarity from word alignment and semantic vector composition. In *Proceedings of SemEval-2015*, pages 148–153, 2015.

[37] E. Westerhout and P. Monachesi. Extraction of Dutch definitory contexts for eLearning purposes. In *Proceedings of the 17th Meeting of Computational Linguistics in the Netherlands*, pages 219–234, 2007.

[38] F. Wu and D. S. Weld. Open information extraction using Wikipedia. In *ACL '10*, pages 118–127, 2010.

[39] J. Xu, R. Weischedel, and A. Licuanan. Evaluation of an extraction-based approach to answering definitional questions. In *SIGIR '04*, pages 418–424, 2004.

[6] https://ciir.cs.umass.edu/downloads/distant_sup_ir/