Expert Systems With Applications 101 (2018) 8–24

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Exploiting semantic similarity for named entity disambiguation in knowledge graphs

Ganggao Zhu*, Carlos A. Iglesias

Universidad Politécnica de Madrid, Escuela Técnica Superior de Ingenieros de Telecomunicación Departamento de Ingeniería de Sistemas Telemáticos, Avenida Complutense 30, Madrid, Spain

ARTICLE INFO

Article history:
Received 14 June 2017
Revised 9 January 2018
Accepted 6 February 2018
Available online 9 February 2018

Keywords:
Entity linking
Named entity disambiguation
Context similarity
Semantic similarity
Word embedding
Knowledge graph

ABSTRACT

With the increasing popularity of large scale Knowledge Graph (KG)s, many applications such as semantic analysis, search and question answering need to link entity mentions in texts to entities in KGs. Because of the polysemy problem in natural language, entity disambiguation is thus a key problem in current research. Existing disambiguation methods have considered entity prominence, context similarity and entity-entity relatedness to discriminate ambiguous entities, which are mainly working on document or paragraph level texts containing rich contextual information, and based on lexical matching for computing context similarity. When meeting short texts containing limited contextual information, such as web queries, questions and tweets, those conventional disambiguation methods are not good at handling single entity mention and measuring context similarity. In order to enhance the performance of disambiguation methods based on context similarity with such short texts, we propose SCSNED method for disambiguation based on semantic similarity between contextual words and informative words of entities in KGs. Specially, we exploit the effectiveness of both knowledge-based and corpus-based semantic similarity methods for entity disambiguation with SCSNED. Moreover, we propose a Category2Vec embedding model based on joint learning of word and category embedding, in order to compute word-category similarity for entity disambiguation. We show the effectiveness of these proposed methods with illustrative examples, and evaluate their effectiveness in a comparative experiment for entity disambiguation in real world web queries, questions and tweets. The experimental results have identified the effectiveness of different semantic similarity methods, and demonstrated the improvement of semantic similarity methods in SCSNED and Category2Vec over the conventional context similarity baseline. We further compare the proposed approaches with the state of the art entity disambiguation systems and show the performances of the proposed approaches are among the best performing systems. In addition, one important feature of the proposed approaches using semantic similarity, is the potential application on any existing KGs since they mainly use common features of entity descriptions and categories. Another contribution of the paper is an updated survey on background of entity disambiguation in KGs and semantic similarity methods.

1. Introduction

The increasing availability of Linked Open Data (LOD) has given birth to the notion of large scale KGs (Bizer, Heath, & Berners-Lee, 2009), with popular examples such as Freebase (Bollacker, Evans, Paritosh, Sturge, & Taylor, 2008), DBpedia (Bizer, Lehmann et al., 2009), and YAGO (Hoffart, Suchanek, Berberich, & Weikum, 2013). Named Entity Linking (NEL) is a fundamental module for developing KG-based applications, including text analysis (Meij, Weerkamp, & de Rijke, 2012), document retrieval (Medelyan, Witten, & Milne, 2008), knowledge base population (Dredze, McNamee, Rao, Gerber, & Finin, 2010; Ji & Grishman, 2011), semantic search and question answering (Shekarpour, Marx, Ngomo, & Auer, 2015). In general, a NEL system needs to detect a sequence of words (spots or mentions) in a given text, and to identify those mentions to entities registered in the given KG. The latter process of entity identification is not a trivial task because it needs to tackle two difficult problems, namely synonymy and polysemy. To address

* Corresponding author.
E-mail addresses: gzhu@dit.upm.es (G. Zhu), cif@dit.upm.es (C.A. Iglesias).

https://doi.org/10.1016/j.eswa.2018.02.011

Table 1
Semantic features of candidate entities for john noble.

Entities	Word features from abstracts	Types
dbr:John_Noble	television, century, theatre, people, winner, actor, director, birth, performance, male, film, horror, fiction, role, series, action	dbo:Actor, yago:Director, dbc:Australian_Film_Actor
dbr:John_Noble(baritone)	baritone, singer, cancer, people, opera, title, favourite, role, composer, progress	yago:Artist, yago:Musician, dbc:English_opera_singers dbc:Operatic_baritones
dbr:John_Noble(bishop)	people, alumnus, bishop, lecturer, school, ministry, career, region, incumbency, teacher, position, chaplain	yago:Bishop, yago:Priest, dbc:Bishops_of_North_Queensland
dbr:John_Noble(painter)	painter, work, carving, landscape, gallon, canvas, photographer, exhibition, outbreak, picture, people, collection, child, post, artist	dbo:Artist, yago:Painter, yago:Creator dbc:20th-century_American_painter

synonymy problem, a NEL system needs to match an entity despite its diverse name variations such as abbreviations, spelling variations, nicknames to name a few. The main approach to solve the synonymy problem is to construct entity name dictionaries as complete as possible in order to cover diverse name variations (Shen, Wang, & Han, 2015), and to apply approximate string matching (Dredze et al., 2010). Thus, the performance of these techniques is mainly concerned with the quality of name dictionaries and approximate matching algorithms. The polysemy problem is caused by the fact that multiple entities in KGs might have the same name, and this is quite common for named entities. The task of addressing the polysemy problem for named entities is called Named Entity Disambiguation (NED), and there is a large body of research techniques that have been proposed for addressing NED automatically (Cucerzan, 2007; Ganea, Ganea, Lucchi, Eickhoff, & Hofmann, 2016; Kulkarni, Singh, Ramakrishnan, & Chakrabarti, 2009; Mendes, Jakob, García-Silva, & Bizer, 2011; Mihalcea & Csomai, 2007; Milne & Witten, 2008). However, resolving the polysemy problem is a common challenge whose difficulty is equivalent to solving central problems of Artificial Intelligence (AI) (Navigli, 2009). The accuracy of NED is far from perfect and related to many aspects of KGs, datasets and applications. This paper focuses on researching of semantic similarity for unsupervised NED using most common semantic features that are available in most KGs, therefore, the proposed similarity-based disambiguation method can be conveniently applied to various KGs.

Current unsupervised NED approaches (Shen et al., 2015) are mainly based on local context (Hoffart et al., 2011), such as context similarity (Mendes et al., 2011), or global inference (Ratinov, Roth, Downey, & Anderson, 2011) using entity-entity relatedness (Milne & Witten, 2008). When the mention contexts are large text objects such as paragraphs or documents, where rich contextual information can be collected, the conventional context similarity approach is effective (Hoffart et al., 2011; Mendes et al., 2011; Mihalcea & Csomai, 2007). However, while processing short texts such as web queries, questions and tweets, limited contextual information may not be effective enough to discriminate ambiguous entities. According to the analysis of commercial search engines (Guo, Xu, Cheng, & Li, 2009), less than 1% of the queries contain two or more named entities, while web queries and questions normally consist of few words (e.g. 3 words on average in search queries and 6–7 words on average in question queries). Obviously, when dealing with limited contextual information, entity-entity relatedness is no longer useful in handling single entity mention and context may not contain enough feature words for computing context similarity. For example, in a web question *"What movies did John Noble play"* (Berant, Chou, Frostig, & Liang, 2013), no other entities can help to discriminate the single ambiguous mention "John Noble". Table 1 shows some candidate entities of mention "John Noble" and their corresponding semantic features extracted from DBpedia (Bizer, Lehmann et al., 2009) which is used as reference KG in this work because of its central role in LOD and various publicly available datasets. The words *movie* and *play* are contextual words, but they do not match the semantic features of the candidate entities. Although the word *movie* is obviously more similar to the entity *dbr:John_Noble* because of the feature words *actor, film, director*, the conventional context similarity can not address such fine-grained semantic closeness to identify the correct entity *dbr:John_Noble*. Consequently, we aim to exploit semantic similarity to develop disambiguation approach that can compare those words in different lexical forms but having similar meanings. In this way, semantic similarity is used to enhance the context similarity for NED when the contextual information is scarce and entity-entity relatedness is not available.

In order to make the similarity-based disambiguation approach applicable to various KGs, entity categories and textual descriptions are used as semantic features to apply semantic similarity methods for NED, because they contain rich semantic information and are available in most KGs. Based on the textual feature, we use Information Retrieval (IR) (Baeza-Yates, Ribeiro-Neto et al., 1999) and Latent Semantic Analysis (LSA) (Deerwester, Dumais, Furnas, Landauer, & Harshman, 1990) to develop the baseline of unsupervised NED approach based on context similarity through the computation of textual similarity between context and entity descriptions. Then we propose a novel Semantic Contextual Similarity based NED (SCSNED), which relies on contextual word similarity to improve the baseline that assumes equal importance of contextual words and provides coarse meaning comparison between context and entity descriptions. The SCSNED computes semantic similarity between individual words to offer fine-grained meaning comparison, and uses inverse entity frequency to consider the relative importance of feature words by counting word appearance in descriptions of candidate entities. In order to optimize the performance of SCSNED, we exploit the usage of both knowledge-based semantic similarity methods (Zhu & Iglesias, 2017a) relying on semantic knowledge of WordNet (Miller, 1995), and corpus-based semantic similarity methods using word embedding model Word2Vec (Mikolov, Chen, Corrado, & Dean, 2013) based on the statistical knowledge from textual corpus. Moreover, given that semantic categories are very effective in representing the meaning of entities (Bekkerman & Gavish, 2011), we propose a Category2Vec embedding model to compute word-category similarity for NED in order to provide complement to the word-word similarity feature. Category2Vec learns semantic category and word embedding jointly based on entity abstracts and entity categories, which treats those categories composed by multi-word expressions (e.g. Australian Film Actor) as a unique semantic unit without separating them into individual words. We found that word-category similarity based on the learned joint vector space is very effective for NED, while the learning of word and category vector representations are only depending on KGs themselves without labeled dataset.

In conclusion, this paper proposes SCSNED method and exploits various similarity methods in the case of little contextual information and single entity mentions. The effectiveness of different similarity methods are identified through a comparative experiment on various datasets including web queries, web questions and tweets. Note that although we evaluate similarity methods with short texts, the proposed NED approach can be directly applied to larger text objects by decomposing them into sentences. Furthermore, we propose a Category2Vec model to compute word-category similarity that has been shown to be effective for NED. The experiments in tweet text have shown that combining baselines, SCSNED and Category2Vec methods have improved the state of art of unsupervised NED approaches. Moreover, unlike many current works optimize effective features for a particular KG, our focus is to exploit some common features and similarity methods that can be used in different kinds of KGs for the task of NED.

The remainder of this paper is structured as follows. In Section 2 we present background of the state of the art NED methods and semantic similarity methods. Section 3 describes SCSNED approach and Category2Vec approach. Our experimental setup and results, as well as detailed analysis are described in Section 4. Conclusions are drawn in Section 5.

2. Background

In this section, we present the overview of NED and describe common semantic similarity methods for words.

2.1. Overview of named entity disambiguation

2.1.1. The definition, scope and related works

Formally, given an input text consisting of a sequence of words $T = \{w_1, w_2, \dots w_k\}$, a NEL system needs to recognize a set of entity mentions $M = \{m_1, m_2, \dots m_n\}$ (called mention detection or entity recognition), and maps each entity mention $m \in M$ to a set of candidate entities E_m which contains all possible entities registered in the given KG that have similar lexical surface form with the entity mention m (called link generation or entity linking). When an entity mention m has more than one entity candidate registered in the KG, $|E_m| > 1$, the NEL system needs to accurately select the correct entity $e \in E_m$ which is the most pertinent to describe the mention m. This process is referred as NED. Since NEL contains both entity recognition and disambiguation, sometimes it is also called Named Entity Recognition and Disambiguation (NERD) (Carmel, Chang, Gabrilovich, Hsu, & Wang, 2014).

Named Entity Recognition (NER) is an important sub-task of Information Extraction (IE) in NLP research for many years. The task of NER is to detect entity mention from unstructured text and determine its categories such as person, location, organization to name a few. Thus, this important sub-task of IE is also called Named Entity Recognition and Classification (NERC) (Nadeau & Sekine, 2007). Many NER approaches have been proposed from early rule-based systems to recent systems employing machine learning techniques (Nadeau & Sekine, 2007). There are many publicly available NER tools such as Stanford NER (Finkel, Grenager, & Manning, 2005) which can be used directly. In many aspects, NER is closely related to NEL because both NER and NEL need to detect entity mention which is the reason of using NER as a precedence for NEL in some works (Hoffart et al., 2011). The main difference between NER and NEL is that NER classifies entity mention into predefined classes while NEL classifies entity mention into entities that are registered in a KB. The predefined classes usually have limited numbers while the number of entities in KG can usually reach to millions so that NEL has the mention-entity mapping process in order to reduce the problem space to a limited number of entity candidates. In addition, the disambiguation task is slightly different. For example, when performing NER in queries (Guo et al., 2009), the disambiguation goal of NER is to classify entity mentions of *Harry Porter* into either class *book* or *movie* according to the given text, while NEL needs to annotate the entity mention with the correct registered entity in KG which can be either a book instance or movie instance. Although the disambiguation goal is different, since entities in KGs normally have a specific entity type (e.g. book or movie), recognizing the entity class can help to determine the corresponding entity in KG. Therefore, recent researches (Guo, Chang, & Kıcıman, 2013; Sil & Yates, 2013) and challenges (Carmel et al., 2014) are proposed to perform NER and NEL jointly. Designing models to represent the relations between entity context (surrounding words) and entity types is one of the main ideas to study NER and NEL jointly. For example, one recent related work (Guo et al., 2009) learns a probabilistic model of the semantic association between entity context and entity type from query log data using Latent Dirichlet Allocation (LDA), where entity context is treated as documents and entity class is treated as topics.

In this work, we do not consider the entity recognition problem and assume that entity mentions are given, which means we only consider the NED instead of the complete NEL system. For more technical details of NEL, of specific KG features, problem scopes, task assumptions, separate tasks[1], technical methods and performances, reader could refer to the complete NEL survey (Shen et al., 2015) and the NEL evaluation (Cornolti, Ferragina, & Ciaramita, 2013; Fernandez Garcia, Arias Fisteus, & Sanchez Fernandez, 2014; Hachey, Radford, Nothman, Honnibal, & Curran, 2013). In addition, those entity mentions having no candidate entities recorded in the given KG, are defined as unlinkable mentions, such as (m_i, *NIL*). Note that NIL is different from *pruning* (Piccinno & Ferragina, 2014) which is used to discard detected mentions and their annotated entities if they are considered not interesting or pertinent to the semantic interpretation of the input text. We do not specially address NIL and *pruning* and assume that all the recognized entity mentions at least have one candidate entity recorded in the given KG. This assumption is similar to some of the works in *Wikification* such as Wikify (Mihalcea & Csomai, 2007).

The task of Word Sense Disambiguation (WSD) (Navigli, 2009) is relevant to NED, because both WSD and NED need to address synonymy and polysemy problem. The task of WSD is defined as automatically assigning the correct sense of a polysemous word within a given context to a given sense inventory such as WordNet (Miller, 1995). For example, a WSD system needs to choose whether the polysemous word *bank* refers to a *repository for money* or a *pile of earth on the edge of a river* within a given context. Because of the similar task shared by NED and WSD, those previously proposed methods for WSD (Gutiérrez, Vázquez, & Montoyo, 2017; Navigli, 2009) are applicable in NED. However, they also meet different challenges. In WSD, words need to be mapped to sense entry (in WordNet, denoted as synset consisting of a set of synonyms) denoting atomic meanings, while named entity mentions need to be mapped to entity entry according to the background KG which are usually multi-words indicating real world things. WSD needs to compare different word meaning, and NED needs to select different entities which are usually different person, place to name a few. In addition, because of the stability of controlled vocabularies, WSD normally assumes that sense inventories such as WordNet are complete where a given word is assumed to be able to find its possible synsets. In contrary, KG are continuously updated (e.g. new entries for new books or movies) and entity mentions are not necessarily mapped to entity entry in the KG (e.g NIL). Furthermore, NED addresses named entities which are modeled as instances in KG, while WSD addresses common nouns (e.g. bank) which are usually treated as metaclasses in KG indicating a group of instance.

[1] https://tac.nist.gov/2017/KBP/.

Therefore, WSD and NED are solving the disambiguation problem in different aspects. Nevertheless, they can be influenced by each other, because instance knowledge can help in class disambiguation while class knowledge is also useful in solving instance disambiguation. This has been shown in many recent researches such as Wikify (Mihalcea & Csomai, 2007) that uses Wikipedia as resource for WSD and Babelfy (Moro, Raganato, & Navigli, 2014) solving WSD and NED jointly based on a wide-coverage semantic network.

Since we evaluate NED approaches with short texts such as web queries, web questions and tweets, we briefly review related works addressing short texts. Query segmentation separates queries into compound words or noun phrases that can be considered as individual concepts (Hagen, Potthast, Stein, & Bräutigam, 2011; Pu & Yu, 2008) aiming to understand the correct query intent for document retrieval. NEL is a component of query understanding (Pound, Hudek, Ilyas, & Weddell, 2012) over KGs for annotating entities in queries for further query classification (Shen, Sun, Yang, & Chen, 2006) or query interpretation (Sawant & Chakrabarti, 2013). Hasibi, Balog, and Bratsberg (2016) exploit the NEL problem with entity retrieval problem jointly in order to improve the search performance. The efficiency problem of linking entities in queries has been studied in (Blanco, Ottaviano, & Meij, 2015) by introducing a probabilistic model, as well as hashing and compression techniques. This paper only studies the effectiveness of NED in case of short texts without employing specific higher level applications. Moreover, NEL has been studied in many research works in case of microblog such as tweets (Guo et al., 2013; Liu et al., 2013; Meij et al., 2012; Shen, Wang, Luo, & Wang, 2013) focusing on processing noisy and informal texts (Guo et al., 2013), user interest model (Shen et al., 2013), entity filtering (Habib & van Keulen, 2015) and empirical analysis of named entity recognition and disambiguation in tweets (Derczynski et al., 2015). We use tweet data as scenario of limited contextual information to evaluate the proposed NED approaches. The state of the art NED approaches are reviewed in the following section, while for techniques of NER in tweets, readers can refer to (Ritter, Clark, Etzioni et al., 2011).

2.1.2. The state of the art

As ambiguous entity mentions have multiple candidate entities, various NED features and methods have been proposed and found to be effective during recent years. To present them clearly, we survey and compare the state of the art NED methods according to a general classification: (1) based on entity prominence; (2) based on context similarity; and (3) based on entity relatedness.

NED methods based on entity prominence select the most prominent entity for a given mention only based on the entity mention and the property of its candidate entities, without considering the surrounding context of the mention. Many prominence features have been used in NED systems including string similarity, popularity and commonness. String similarity is based on the name string comparison between mention and candidate entities using different similarity or distance metrics such as edit distance (Liu et al., 2013), Dice, Hamming distance (Dredze et al., 2010) to name a few. String similarity is the most straightforward and common feature for NED, but it is not reliable when candidate entities have same names or the mentions have many name variations. Popularity is another common prominence feature which is domain dependent and especially useful in the case of lacking contextual information, such as single entity mention in the query. Wikipedia page view statistics is a typical popularity feature that has been used to represent the entity popularity in many systems (Gattani et al., 2013; Guo et al., 2013). Another popularity feature is based on click popularity (Ji & Grishman, 2011). Both page view and click information are effective to select the most popular entity for individual ambiguous entity mentions when there is no other information to help discriminate candidate entities. However, both page view and click features are dependent on domain specific applications and will meet cold-start problem. Moreover, commonness (Medelyan et al., 2008) has been proven as a very effective prominence feature in many NED systems (Ferragina & Scaiella, 2010; Guo et al., 2013; Hoffart et al., 2011; Kulkarni et al., 2009; Liu et al., 2013; Medelyan et al., 2008; Ratinov et al., 2011; Shen, Wang, Luo, & Wang, 2012b). The entity commonness denotes the prior probability of entity which is computed from sense distribution over entity annotation corpora such as anchor text of Wikipedia. If a word or n-gram a appears as an annotation in corpora N times and there are m times linking to the entity E, then the *commonness* of entity E can be computed as $P(E|a) = \frac{m}{N}$. Computing entity commonness is dependent on entity annotation corpora which is difficult to obtain and the computed entity commonness probability may only have limited entity coverage because of the incompleteness of the annotation corpora.

NED methods based on context similarity discriminate ambiguous entities through measuring similarity between the mention context and the candidate entities. Context similarity metrics depend on different semantic features in representing contexts and entities. The most intuitive semantic features to represent context are different granularity of texts surrounding the mention, from whole input text to several surrounding words. Similarly, entities can be represented by the textual descriptions extracted from KGs, ranging from the entire Wikipedia page (Bunescu & Pasca, 2006), paragraphs of Wikipedia page (Kulkarni et al., 2009; Mendes et al., 2011), entity summaries (Ratinov et al., 2011), entity abstracts (Meij, Bron, Hollink, Huurnink, & de Rijke, 2011), entity categories (Bunescu & Pasca, 2006), entity types (Guo et al., 2013), keyphrases (Hoffart et al., 2011), entity titles (Liu et al., 2013), and anchor texts (Kulkarni et al., 2009) to name a few. Then, context-entity similarity can be computed with different similarity metrics based on different models for textual features. The simplest model is Bag of Words (BOW) where both contexts and entities are represented as set of words, concepts or keyphrases (Hoffart et al., 2011). In consequence, the context-entity similarity is computed based on set intersection that counts the overlap of words, concepts, keyphrases (Hoffart et al., 2011) between context and entity (Mihalcea & Csomai, 2007) similar to the idea of Lesk algorithm (Lesk, 1986) for WSD. Common metrics to compute such overlap similarity are Jaccard similarity or Dice coefficient (Bunescu & Pasca, 2006; Hoffart et al., 2011; Kulkarni et al., 2009; Liu et al., 2013; Mihalcea & Csomai, 2007). Apart from the simple BOW, Vector Space Model (VSM) has been used to represent contexts and entities into high dimensional context and entity vectors, whose values of each dimension are Term Frequency (TF)-Inverse Document Frequency (IDF) scores (Baeza-Yates et al., 1999) computed from specific text collection and particular vocabulary. Then the context-entity similarity is computed using dot-product or cosine similarity between context and entity vectors (Bunescu & Pasca, 2006; Dredze et al., 2010; Guo et al., 2013; Han, Sun, & Zhao, 2011; Kulkarni et al., 2009; Mendes et al., 2011; Milne & Witten, 2013; Ratinov et al., 2011). Moreover, some recent works proposed to learn distributed vector representation of mention, context and entity for NED (Francis-Landau, Durrett, & Klein, 2016; He et al., 2013; Sun et al., 2015) with deep learning architecture (Hinton, Osindero, & Teh, 2006), and proposed to extend the contextual words with similar words (Blanco et al., 2015) based on Word2Vec (Mikolov et al., 2013). In addition, probabilistic language models (Han & Sun, 2011; Meij et al., 2011) and topic models (Houlsby & Ciaramita, 2014; Kataria, Kumar, Rastogi, Sen, & Sengamedu, 2011; Pilz & Paaß, 2011) have been applied to model context, mention and candidate entity in order to rank the entities given the specific context and mention. The context and entity co-occurrence knowledge are encoded to compute context-entity similarity according to the probabilistic likelihood of an entity appearing in a specific context. Following such idea, recent NED works formulate probabilistic framework considering various type of statistical information including mention-entity probability, entity-entity co-occurrence, contextual word-entity statistics from entity annotation dataset (e.g. Wikipedia anchor dataset) (Blanco et al., 2015; Ganea et al., 2016).

Entity relatedness is a special case of context similarity, since entities of other mentions in the input text are used as the semantic feature to represent context. According to the assumption that the input text contains coherent entities from one or few related topics (Hoffart et al., 2011), multiple ambiguous entities are discriminated collectively (Kulkarni et al., 2009; Zhang, Rettinger, & Philipp, 2016) based on entity relatedness. Such collective disambiguation model is a global model that discriminates all entity mentions jointly (Ratinov et al., 2011). In contrast, NED methods based on entity prominence and context similarity use frequently a local model (Ratinov et al., 2011) which considers each entity mention in isolation. Collective disambiguation (Hoffart et al., 2011) of all entity mentions in an input text is shown as NP-hard problem, which is usually simplified by comparing unambiguous entities with ambiguous entities (Cucerzan, 2007), combining with entity prominence (Ferragina & Scaiella, 2010), or averaging the coherence (Shen et al., 2012b). In fact, most of NED methods combine both local and global models to achieve better disambiguation performance (Ratinov et al., 2011). The key module of this collective disambiguation model is measuring entity relatedness in order to infer the coherence among candidate entities for all mentions. There is a number of semantic features that can be used to compute entity-entity relatedness based on different type of information sources. Firstly, semantic contents of entities such as textual descriptions and semantic categories are represented in BOW or VSM to compute entity-entity similarity based on: (1) dot or cosine similarity of entity description or category vectors (Cucerzan, 2007); (2) topical coherence between entities using overlap of weighted keyphrases (Hoffart, Seufert, Nguyen, Theobald, & Weikum, 2012) and topic models (Piccinno & Ferragina, 2014); and (3) semantic similarity of entity category hierarchies (Shen et al., 2012b). Secondly, from entity annotated corpora, entity co-occurrence (Nunes et al., 2013) and entity distribution (Aggarwal, Asooja, Ziad, & Buitelaar, 2015; Shen, Wang, Luo, & Wang, 2012a) are used to compute entity-entity relatedness based on the application of distributional hypothesis (Turney, Pantel et al., 2010) which assumes that entities occur in similar contexts are semantically related. Finally, apart from semantic content analysis and distributional analysis, graph analysis is also very effective in measuring entity connectivity in order to compute entity-entity relatedness, given that entities are connected to each other in KGs. Graph analysis measures the entity relatedness based on semantic entity networks using degree analysis (Milne & Witten, 2008) or relational analysis (Hulpuş, Prangnawarat, & Hayes, 2015). Degree analysis counts the edges connecting entities which only represent occurrence, incoming, or outgoing information, while relational analysis considers semantic meaningful relations between entities. This difference results in different kind of entity relatedness methods. Milne and Witten (2008) proposed a degree analysis method for computing entity relatedness based on the incoming and outgoing links, which is similar to the Normalized Google Distance (Cilibrasi & Vitanyi, 2007). This entity relatedness method has been popularly adopted by many subsequent NED systems (Ferragina & Scaiella, 2012; Han et al., 2011; Han & Zhao, 2009; Hoffart et al., 2011; Kulkarni et al., 2009; Liu et al., 2013; Medelyan et al., 2008; Ratinov et al., 2011; Shen et al., 2012b). Following a similar idea, some variations based on degree analysis include Pointwise Mutual Information (Ratinov et al., 2011) and Jaccard distance (Guo et al., 2013). Recent works started to consider semantic relations between entities in KG and use relation analysis for computing entity relatedness based on the shortest path between entities (Nunes et al., 2013) and relation weighting in the shortest path (Hulpuş et al., 2015).

With various entity relatedness methods, it is obvious that the performance of entity relatedness can be further optimized and enhanced by combining different methods through supervised machine learning techniques (Ceccarelli, Lucchese, Orlando, Perego, & Trani, 2013). Similarly, better NED performance can be achieved by combining different disambiguation methods and features, and using a labeled dataset to learn to assign proper entities with supervised learning methods, such as naive bayes classier (Mihalcea & Csomai, 2007), support vector machine (Bunescu & Pasca, 2006; Meij et al., 2011; Ratinov et al., 2011), and learning to rank framework (Milne & Witten, 2008) to name a few. As preparing labeled datasets requires tremendous efforts, when labeled datasets are not available, unsupervised NED approaches are needed. A simple but effective unsupervised disambiguation method is to select the correct entity with the highest similarity score computed based on entity prominence, context similarity and entity relatedness (Ferragina & Scaiella, 2010; Han & Sun, 2011; Medelyan et al., 2008; Mendes et al., 2011; Shen et al., 2012b). Apart from similarity-based methods, more complicated unsupervised disambiguation method is graph-based approach that combines various disambiguation features into graph representation. Specifically, mention, context and entity are modeled as nodes in graph, while their semantic associations are modeled as edges. Then, various graph-based algorithms (Piccinno & Ferragina, 2014) can be applied to make disambiguation decisions, such as PageRank (Hachey, Radford, & Curran, 2011), Personalised PageRank (Agirre & Soroa, 2009; Han et al., 2011), dense subgraph estimation (Hoffart et al., 2011), degree-based importance measure (Guo, Che, Liu, & Li, 2011), n-cliques model (Alhelbawy & Gaizauskas, 2014; Gutiérrez, Vázquez, & Montoyo, 2012), Hypertext-Induced Topic Search (HITS) (Usbeck, Ngonga Ngomo, Auer, Gerber, & Both, 2014) and many others. Various disambiguation features and methods described above represent different aspects and consideration in dealing NED. No feature or method is superior than others over all kinds of datasets (Shen et al., 2015). Thus, disambiguation features and methods need to be selected according to the specific characteristic of dataset and the requirement of application in tradeoff between precision and recall, accuracy and efficiency.

This paper focuses on the NED methods based on context similarity. Instead of computing context-entity similarity based on text-text or entity-entity similarity, we propose unsupervised NED methods based on word-word, and word-category semantic similarity to enhance the performance of context-entity similarity computation with the fine-grained meaning comparison.

2.2. Overview of semantic similarity

Semantic similarity methods give numerical similarity scores to words in order to represent their semantic distance. In computational linguistics, semantic relatedness is inverse of semantic distances and assumes that two objects are semantically related if they have any kind of semantic relations (Budanitsky & Hirst, 2001). Semantic similarity is a special metric that represents the commonality of two concepts relying on their hierarchical relations (Turney et al., 2010). In general, semantic similarity is a special case of semantic relatedness (Turney et al., 2010) which is a more general concept and does not necessarily rely on hierarchical relations. This work covers both semantic similarity and semantic relatedness. For convenience we call them semantic similarity interchangeably in the following sections and categorize them into corpus-based methods and knowledge-based methods (Zhu & Iglesias, 2017a). Corpus-based methods mainly rely on contextual information of words appearing in the corpus, thus they mainly measure general semantic relatedness between words. Knowledge-based methods derive semantic similarity of words based on hierarchical relations encoded in WordNet. Corpus-based methods have wider computational applications because they consider all kinds of semantic relations between words, while knowledge-based methods would be more useful when applications need to encode hierarchical relations between words. We review both types of methods in the following sections.

2.2.1. Corpus-based methods

Corpus-based semantic similarity methods are based on word associations learned from large text collections following the distributional hypothesis (Turney et al., 2010). Two words are assumed to be more similar if their surrounding contexts are more similar or they appear together more frequently. The computation of corpus-based methods are based on statistics of word distributions or word co-occurrences. According to different computational models, there are count-based methods, e.g. Pointwise Mutual Information (Church & Hanks, 1990) or Normalized Google Distance (Gligorov, ten Kate, Aleksovski, & van Harmelen, 2007), and predictive methods, e.g. Word2Vec (Mikolov et al., 2013). Count-based methods count word co-occurrences and construct a word-word matrix, in which those co-occurrence statistics are directly applied with probabilistic models (Church & Hanks, 1990), matrix factorization (Pennington, Socher, & Manning, 2014) and dimension reduction (Levy, Goldberg, & Dagan, 2015). Predictive-based methods directly learn dense vectors through predicting a word from its surrounding context. We use the predictive-based word embedding tool Word2Vec (Mikolov et al., 2013) to learn dense vector representation of words, because it has been reported to have good performance in many applications (Baroni, Dinu, & Kruszewski, 2014) and our proposed Category2Vec model is based on it. As suggested by the Word2Vec authors (Mikolov et al., 2013), the Continuous Bag of Words (CBOW) model is more computationally efficient and suitable for larger corpus than the skip-gram model. Thus, the CBOW model is used to train word vectors in a neural network architecture which consists of an input layer, a projection layer, and an output layer to predict a word given its surrounding words with a certain context window size. Formally, given a sequence of training words $\{w_1, w_2, \dots w_T\}$, each word vector is trained to maximize the average log probability:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq k\leq c, k\neq 0} logp(w_t|w_{t-k}, \dots, w_{t+k}) \tag{1}$$

where k is the context window size and $p(w_t|w_{t-k}, \dots, w_{t+k})$ is the hierarchical softmax of the word vectors (Mikolov et al., 2013). Having the trained word vectors (the dimension is predefined empirically and we set 300 in our experiments), word similarity are computed using standard cosine similarity. Although the training process relies on a neural network based supervised prediction model, the real training results are the vector representation of words instead of the neural network prediction model. Because of such idea, the training of word embedding is unsupervised and can be applied in various textual corpus without labeled dataset, which makes Word2Vec applicable to most KGs containing textual descriptions. Furthermore, due to the simple neural network architecture and the use of hierarchical softmax, Word2Vec is able to address large corpus and the training is very efficient. However, since the training of word vectors only use word sequences, a wide variety of word relations are considered as equally related according their co-occurrences, which makes the similarity between trained word vectors coarse and unable to address synonymous words and hierarchical relations accurately. In consequence, knowledge-based semantic similarity methods are considered to enrich some commonsense knowledge of words.

2.2.2. Knowledge-based methods

Knowledge-based semantic similarity methods measure the semantic similarity between words based on an ontology. Two words are considered to be more similar if they are located closer in the given ontology. The lexical database WordNet (Miller, 1995) is used as background ontology, which is organized through synsets, being each synset a set of words sharing one common sense (synonyms). The hierarchical relations between synsets (i.e. hypernym and hyponymy), organize WordNet into a concept taxonomy. Having synonymous words in synsets and human defined hierarchical relations, knowledge-based semantic similarity methods are designed to encode this information to improve semantic similarity between words. Those semantic similarity methods are directly used for synsets rather than words, therefore, they need to be converted into word similarity by taking the maximal similarity score over all the synsets which are the senses of the words (Resnik, 1995; Sánchez, Batet, Isern, & Valls, 2012). This is based on the intuition that human would pay more attention to word similarities (i.e., most related senses) rather than their differences (Sánchez et al., 2012), which has been demonstrated in psychological studies (Tversky, 1977). As polysemous words can be mapped to a set of synsets, let $s(w)$ denote a set of synsets that are senses of word w, then word similarity is defined as:

$$sim_{word}(w_i, w_j) = \max_{c_i\in s(w_i), c_j\in s(w_j)} sim_{synset}(c_i, c_j) \tag{2}$$

where sim_{synset} can be any semantic similarity methods used for WordNet synsets which are presented in the following of this section.

Many knowledge-based methods have been proposed in the literature (Budanitsky & Hirst, 2006) for measuring similarity in WordNet exploiting various information such as shortest path length, depth, and Information Content (IC). The basic idea is counting the number of nodes or edges (shortest path) between two concepts (synsets) in WordNet. Two concepts are assumed to be more similar if they are closer to each other in WordNet. Let $path(c_i, c_j)$ be the shortest path length between c_i and c_j, the Path (Rada, Mili, Bicknell, & Blettner, 1989) method defines semantic similarity method as:

$$sim_{Path}(c_i, c_j) = \frac{1}{1 + path(c_i, c_j)} \tag{3}$$

Another common information used to compute semantic similarity is depth, which is defined as shortest path length between root concept and a given concept through hierarchical relations. The intuition behind depth is that the upper-level concepts in a taxonomy are supposed to be more general. Thus, the similarity between lower-level concepts should be considered more similar than those concepts between upper-level concepts. The Wu and Palmer (1994) method measures the semantic similarity between concepts based on concept depth in a taxonomy, and a special concept Least Common Subsumer (LCS), which is the most specific ancestor concept shared by two concepts. Let $LCS(c_i, c_j)$ be the LCS of concepts c_i and c_j, the Wu and Palmer (1994) method measures semantic similarity of given concepts using the following formula:

$$sim_{Wu\&Palmer}(c_i, c_j) = \frac{2depth(LCS(c_i, c_j))}{depth(c_i) + depth(c_j)} \tag{4}$$

where $depth(c_i)$ computes the $path(c_{root}, c_i)$ given c_{root} is the root concept of the taxonomy. The two similarity methods described above consider the structural knowledge of a taxonomy which have a common drawback of uniform distance between concepts. Some methods consider IC to overcome the uniform distance drawback. The IC is defined as the probability of encountering the concept in a corpus $IC(c_i) = -logProb(c_i)$. Note that Brown Corpus (Francis & Kucera, 1979) is used to compute IC because words in BC are annotated with WordNet concepts. The Resnik (1995) method only considers the IC of LCS concept, while the consequent works by Lin (1998) and Jiang and Conrath (1997) extend the IC-based method by including the IC of concepts.

$$sim_{Resnik}(c_i, c_j) = IC(LCS(c_i, c_j)) \tag{5}$$

$$sim_{Lin}(w_i, w_j) = \frac{2IC(LCS(c_i, c_j))}{IC(c_i) + IC(c_j)} \tag{6}$$

$$sim_{Jiang\&Conrad}(w_i, w_j) = \frac{1}{1 + IC(c_i) + IC(c_j) - 2IC(LCS(c_i, c_j))} \tag{7}$$

Note that Eq. (7) transforms original semantic distance into similarity and solves the divide by zero problem. As IC-based methods are lack of important information of path and depth, they are not able to represent concept's distance and specificity accurately. WPath (Zhu & Iglesias, 2017a) combines structural knowledge and statistical IC to have hybrid semantic representation between concepts.

$$sim_{wpath}(c_i, c_j) = \frac{1}{1 + path(c_i, c_j) * k^{LCS(c_i, c_j)}} \tag{8}$$

where $k \in (0, 1]$ and $k = 1$ means that IC has no contribution in shortest path length. The parameter k ($k = 0.8$ is used as the original proposal) represents the contribution of the LCS's IC which indicates the common information shared by two concepts. WPath aims to give different weights of the shortest path length between concepts based on their shared information, where the path length is viewed as difference and the IC is viewed as commonality. For identical concepts, their path length is 0 so their semantic similarity reaches the maximum similarity 1.

Apart from semantic similarity methods for specific ontology such as WordNet, recent works also started to propose semantic distance and similarity methods for LOD, where wide coverage of semantic relations between semantic resources are provided. Passant (2010) introduced a set of semantic distance metrics based on different links between resources and applied those metrics to cases of resource recommendation. REWOrD (Pirró, 2012) computes relatedness between entities based on weighted vectors which are constructed from informative predicates of entities. Moreover, recent work of Meymandpour and Davis (2016) made a survey on state of the art of semantic similarity and its application in terms of LOD, and presents an information content-based approach to compute semantic similarity between entities considering the relative importance of various types of entity features available in LOD. Shi, Yang, and Weninger (2017) proposed a dual perspective similarity metric that calculates the similarity between nodes in the network based on the perspective of both the query node and the candidate endpoint, whose effectiveness has been validated in scenarios such as community analysis and link prediction. These similarity methods are proposed for more general semantic network and focused on entity level resources. This paper discusses semantic similarity between common nouns based on WordNet since those common words represent conceptual abstractions of entities such as singer, actor to name a few. We mainly compare the different impact of different word semantic similarity methods for the entity disambiguation problem.

2.2.3. Illustrative comparison

Corpus-based and knowledge-based methods have different pros and cons in measuring word similarity. Corpus-based methods usually have better coverage of vocabulary because their computational models can be effectively applied to various and updated corpora. In case of KGs, since many entity descriptions normally contain domain specific terms which are not covered in common sense dictionaries such as WordNet, corpus-based tools like Word2Vec can capture domain specific vocabulary. On the other hand, because corpus-based methods do not consider different word meanings and various word relations, the learned word vectors are not as accurate as knowledge-based methods in some cases when words have special relations. For example, as illustrated in Table 2, *movie* and *film* are synonyms so they have highest similarity score of one in knowledge-based methods, while *baritone* is a sub-concept of *singer* so they should be more similar than *actor* and *singer*. Furthermore, since the main semantic information used by corpus-based methods are word sequence statistics from corpora, when the training corpora change, the word vectors would change and the similarity between words are different. While knowledge-based methods rely on ontologies which are normally fixed and stable, word similarity scores are different only when the corresponding similarity metric changes. In Table 2, we have shown the WPath (Zhu & Iglesias, 2017a) method computing word similarity based on WordNet, and Word2Vec model trained from Wikipedia dump. We will compare different knowledge-based methods and Word2Vec model with different corpora in our experiment.

In addition, comparing rows and columns of Table 2, both types of similarity methods have given the same rank orders to those word pairs through their similarity scores, whereas some cases also show the difference between two kinds of methods. Two main reasons might have caused such a difference. Firstly, knowledge-based methods mainly study the concept taxonomy of WordNet, thus they are preferred to give higher similarity to those concepts in the same branch of the taxonomy, and give lower similarity to related words, such as *actor* and *movie*. Secondly, many common sense knowledge is usually not described so it is not contained in many textual corpus. In this case, corpus-based method may not be able to represent it. For example, there are some zero similarity values in corpus-based methods and the word *play* is given low similarity to *film* and *actor*. In summary, considering those pros and cons of both types of methods, it is better to combine both for NED in a given domain.

3. Semantic contextual similarity for NED

In this section, we present baseline approaches, and propose SCSNED and Category2Vec approaches for unsupervised NED based on semantic contextual similarity.

3.1. The baseline approaches

As entity descriptions are common and effective textual features available for most of KGs, IR techniques (Baeza-Yates et al., 1999) have been applied in many NED systems (Bunescu & Pasca, 2006; Mendes et al., 2011; Mihalcea & Csomai, 2007) to compute text similarity scores to discriminate the ambiguous candidates. We use them as the baseline approach of NED based on context similarity.

Context similarity is based on measuring vector similarity over standard VSM for mention context and entity descriptions, where both contexts and entities are represented as high dimensional vectors $v \in \mathbb{R}^{|V|}$. Each dimension of the vector v corresponds to a word in the vocabulary V which is created from all the entity descriptions in KGs. When the vocabulary is created, lemmatization is applied and those stop words, too frequent words and too rare words are filtered based on application requirements. In our illustrative example of DBpedia (see Table 1), words that appear less than 20 times and occur in descriptions of more than 50% entities have been removed, which has resulted in a vocabulary $|V| = 100000$. The value in each dimension of vector v is represented by the corresponding word weight and computed using standard TF and IDF (Baeza-Yates et al., 1999). Formally, $tf(w_i, d)$

Table 2
Word similarity comparison between knowledge-based method WPath (Zhu & Iglesias, 2017a) and Corpus-based method Word2Vec (Mikolov et al., 2013).

Words	Knowledge-based								Corpus-based							
	Movie	Play	Singer	Teacher	Painter	Composer	Opera	Theatre	Movie	Play	Singer	Teacher	Painter	Composer	Opera	Theatre
film	1.000	0.660	0.160	0.143	0.295	0.143	0.218	0.251	0.751	0.170	0.224	0.079	0.105	0.236	0.179	0.310
actor	0.150	0.125	0.544	0.252	0.296	0.252	0.135	0.160	0.401	0.168	0.550	0.246	0.356	0.423	0.236	0.314
baritone	0.157	0.230	0.839	0.158	0.174	0.158	0.204	0.157	0.054	0.147	0.450	0.224	0.259	0.497	0.42	0.123
director	0.113	0.010	0.416	0.174	0.587	0.728	0.105	0.118	0.200	0.037	0.225	0.308	0.150	0.345	0.131	0.265
bishop	0.157	0.251	0.174	0.158	0.174	0.158	0.143	0.157	0.000	0.032	0.083	0.205	0.124	0.154	0.062	0.058
picture	1.000	0.660	0.68	0.123	0.251	0.113	0.230	0.218	0.407	0.085	0.072	0.050	0.142	0.066	0.029	0.142
photographer	0.123	0.100	0.471	0.194	0.681	0.587	0.113	0.130	0.180	0.000	0.387	0.381	0.567	0.440	0.082	0.123

denotes the frequency of word w_i in the document d, while $df(w_i)$ denotes the document frequency of the word w_i which is numbers of entities whose textual descriptions contain the word w_i. The weight of dimension i of document d is defined as the product of TF and IDF of word w_i:

$$v_{i,d} = tf(w_i, d) * (1 + log\frac{N}{1 + df(w_i)}), \tag{9}$$

where N is the total number of entities in KG and word w_i corresponds to a token in the vocabulary V. If the word w_i in vocabulary V is not contained in a particular document d, then $tf(w_i, d) = 0$ and $v_{i,d} = 0$. Given a context vector v_c and an entity description vector v_{e_i}, the cosine similarity is the cosine of the angle between vectors of context and entity:

$$sim_{cos}(v_c, v_{e_i}) = \frac{v_c \cdot v_{e_i}}{\|v_c\|_2 \times \|v_{e_i}\|_2} \tag{10}$$

The cosine similarity between context vector v_c and entity vector v_{e_i} can be viewed as the degree of correlation between words from mention context and entity description. Since mention context and candidate entities usually only contain a few words from vocabulary, the vector v is normally a sparse vector so the correlation may be very low when entity descriptions do not contain many words appearing in the mention context. The vocabulary mismatch problem would result in failure of measuring similarity between context and entities when they do not contain same words in the vocabulary. Furthermore, the construction of vector v only counts those words appearing in contexts or entities without considering their related words semantically.

In order to overcome the sparseness and vocabulary mismatch problem in standard IR-based text similarity model, a topic model LSA (Deerwester et al., 1990) is used to group similar words into latent topics through dimension reduction. Higher dimensional TF-IDF vectors are transformed into lower dimensional dense vectors in which each dimension denotes a latent topic. With those latent topics, even if the contextual words are not occurred in the description of candidate entities, the occurrence of their synonyms or related words can be counted as meaningful evidence to indicate the semantic relevance between contexts and entities. This is achieved by measuring similarity between context and entities in a second order relations among words. LSA operates Singular Value Decomposition (SVD) on the TF-IDF word-entity matrix M of vocabulary V and N entity descriptions from KG. The semantic representation is obtained from word co-occurrence information by discovering latent topics and using them to represent contexts and entities. Formally, SVD factors the matrix M into three matrices according to the following equation:

$$M_{|V|\times N} = U_{|V|\times K} \Sigma_{K\times K} S^T_{K\times N} \tag{11}$$

where $\Sigma_{K\times K}$ is the diagonal $K \times K$ matrix containing the K singular values and U and S are orthogonal matrices. $|V|$ and N are the number of words and entities. Typically we can remove some insignificant dimensions by retaining only the K' largest singular values in Σ and setting the remaining small ones to zero. The original M is approximated by K' largest singular triplets and the new vector space becomes the latent semantic topical vector space. In consequence, the original context and entity vector v in standard VSM can be transformed into K' dimensional topic vectors through the following equation:

$$\hat{v} = v^T U_{|V|\times K'} \Sigma^{-1}_{K'\times K'} \tag{12}$$

With the lower dimensional topic vectors of context and candidate entities ($K' = 300$ is chosen in this work), the context similarity is also implemented through cosine similarity defined in Eq. (10).

In summary, LSA creates a vector space model with latent topics rather than vocabulary V, and enables a homogeneous representation of words, sentences and documents. Because of this, LSA is able to address vocabulary mismatch problem between contextual words and entity descriptions, and give meaningful similarity scores to rank the candidate entities. For an illustrative example of entity mention *John Noble*, Table 3 shows the text similarity scores between contextual word *movie* and its candidate entities based on standard IR and LSA respectively. The example shows that the TF-IDF has failed in ranking candidate entities since it gives zero score to every candidate when *movie* has not occurred in all the entity descriptions. In terms of LSA, since word *movie* and entity descriptions are mapped into latent topical vector space, all the candidates have been assigned a similarity score, while the expected entity *dbr:John_Noble* has been given higher text similarity score than other entities. The example demonstrates that LSA handles better the vocabulary mismatch over TF-IDF. In addition, both IR and LSA model the context-entity similarity using text similarity, which has coarse-grained semantic meaning representation since text vectors are composed from multiple words. In the following sections, we present a fine-grained meaning representation using semantic similarity between words.

3.2. The SCSNED approach

When contextual information is limited, contextual words are key evidences for NED, therefore more fine-grained semantic similarity models are critical to quantify the relevance between context and candidate entities through word similarity.

Semantic association between contextual words and candidate entities can help to select the proper entity. Since both textual descriptions and categorical labels of entities contain informative words to represent candidate entities, the word-entity similarity can be measured through word-word similarity between contextual words and entity feature words. For example, the entity *dbr:John_Noble* has the textual description "John Noble (born 20 August 1948) is an Australian film and television actor...", and categorical labels from Wikipedia categories *1948 births Australian male film actors*. Meaningful feature words can be extracted from those textual descriptions. For illustration, the example of extracted feature words of candidate entities for the mention *John Noble* are shown in Table 1. Note that nouns are usually more informative than verbs, adjectives and adverbs for NED so we mainly consider

Table 3
Context-entity similarity based on TFIDF or LSA.

Model	dbr:John_Noble	dbr:John_Noble(baritone)	dbr:John_Noble(bishop)	dbr:John_Noble(painter)
TFIDF	0.0	0.0	0.0	0.0
LSA	0.355	0.094	0.0075	0.0874

nouns as feature words of an entity. Words such as *actor, film, director* in the entity *dbr:John_Noble* are more relevant to the word *movie* than other words in other entities such as *baritone, singer, bishop, school, painter, photographer*.

Algorithm 1 outlines the details of SCSNED approach for disambiguation. Given the mention context and a set of entity candidates, the approach tries to identify the correct entity for the given mention. The functions *words*(*context*) and *words*(*entity*) retrieve the feature words for context and entity respectively. In order to quantify this similarity model between words, a word similarity function $sim_{word}(w_i, w_j) \in [0, 1]$ is used to give numerical score of the similarity between word w_i and w_j. Formally, BOW is used to represent both contexts and candidate entities while semantic similarity is used to compare individual items in two sets semantically, instead of lexical matching. Given a set of entity candidates $E = \{e_1, \ldots, e_n\}$ for mention *m*, $F(e_i)$ receives the feature words of a candidate entity e_i and *C* receives a set of feature words in the surrounding context of mention *m*. Then the word similarity scores between words in context and words in candidate are computed. The top *K* similarity scores are selected and summed to generate a weight value as being the correct entity, which is shown as $value \leftarrow sum(top(S, K))$ in Algorithm 1. This similarity computation is repeated over all the candidate entities, while the entity with highest weight value is returned as correct entity. The formal definition of SCSNED approach is shown in the following function:

Algorithm 1 The SCSNED approach for disambiguation.

```
procedure DISAMBIGUATE(context, candidates, K)
    $C \leftarrow words(context)$
    $E \leftarrow candidates$
    $ef \leftarrow frequency(word \in candidates)$
    $score \leftarrow 0, entity \leftarrow \emptyset$
    for all $e \in E$ do
        $F(e) \leftarrow words(e)$
        for all $w \in C$ do
            for all $f \in F(e)$ do
                $S \leftarrow sim_{word}(w, f) * (1 + log\frac{|E|}{1+ef(f)})$
            end for
        end for
        $value \leftarrow sum(top(S, K))$
        if $value > score$ then
            $entity \leftarrow e$
            $score \leftarrow value$
        end if
    end for
    return $entity$
end procedure
```

$$\hat{e} = \underset{e_i \in E}{argmax} \sum_{w_i \in C, w_j \in F(e_i)}^{K} sim_{word}(w_i, w_j) * \left(1 + log\frac{|E|}{1 + ef(w_j)}\right) \tag{13}$$

where |*E*| is numbers of candidate entities and $ef(w_j)$ counts numbers of candidate entities containing word w_j. $\sum^K$ is a sum function that selects and sums the top-K word similarity scores. The idea of $ef(w_j)$ is similar to document frequency for computing IDF. Given that some feature words may occur frequently in entity descriptions (e.g. word *person* occurs in every candidate entity of *john noble*), $ef(w_j)$ is used to give lower weight to those less discriminative words. *K* is designed as parameter that can be determined empirically or optimized from datasets. If *K* is set to smaller values such as one, two or three, the SCSNED may not able to discriminate multiple candidate entities because they may contain several feature words having same similarity scores. On the other hand, if *K* has been set a bigger value, too much irrelevant feature words having lower similarity scores may be included in ranking so that the ranking precision would be affected. The value of K can be defined empirically for each dataset, to find a trade-off among too many irrelevant words (high K value) and too few feature words (low K value). The value for this paper has been set to 10 to test the performance of the different similarity methods.

Furthermore, the SCSNED becomes a semantic ranking model relying on semantic similarity of meaningful words, which is more accurate than text similarity in addressing synonymous and polysemous words. Since users in different backgrounds will describe the same information using different words (e.g. *movie* and *film* are synonymous words), semantic similarity can solve this problem by giving higher similarity scores to those semantically equivalent but lexically different words. Moreover, semantic similarity can partially solve polysemous word problem since it always gives highest similarity score between two words representing their closest meaning. Because of the important role of semantic similarity methods for words, we exploit the usage of both corpus-based and knowledge-based semantic similarity methods for the SCSNED approach in evaluation.

3.3. The Category2Vec approach

According to near-sufficiency property (Bekkerman & Gavish, 2011), semantic categories are informative to represent the meaning of an entity (e.g. director, actor) which usually contain more information about entity than longer entity descriptions. For example, as shown in Table 1, a candidate entity *dbr:John_Noble* has semantic categories such as yago:Director, dbc:Australian_Film_Actor. In the previous section, we have discussed the word similarity method based on meaningful words extracted from decomposing categories into individual words. The decomposition process may lose specific meaning of multi-word expressions. For example, *dbc:Australian_Film_Actor* has a more specific meaning indicating a specific group of actors, than those individual words of Australian, film and actor. Moreover, many current works (Bunescu & Pasca, 2006; Cucerzan, 2007; Shen et al., 2012b) have studied semantic categories such as Wikipedia categories for entity disambiguation. However, to the best of our knowledge, current works mainly focus on category to category similarity (Bunescu & Pasca, 2006; Shen et al., 2012b) for disambiguation (Kulkarni et al., 2009), which would fail in single entity mention. Considering those problems in using semantic categories, we aim to develop word-category similarity to be the complement of word-word model in order to retain the complete meaning of categories.

Semantic categories can be viewed as semantic tags annotating entity descriptions, providing meaningful abstract keywords to entity descriptions. The co-occurrence of categories and

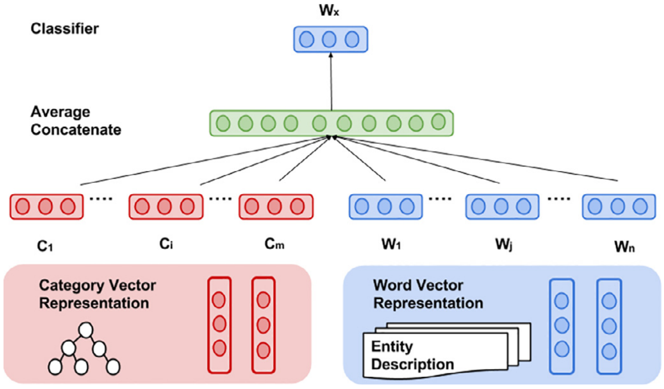

Fig. 1. Jointly learning embeddings of word and category through entities in KG.

words in entity descriptions can be used to learn the word-category associations. Applying the distributional semantics hypothesis (Turney et al., 2010), a word and a category are assumed to be more similar if they appear together more frequently. Following this idea, Word2Vec (Mikolov et al., 2013) can be used to learn words and categories embedding by treating categories as special tokens appearing together with words. Then the similarity between word and category can be computed by cosine similarity of their vectors in shared vector space. We use Doc2Vec (Le & Mikolov, 2014) to build Category2Vec model for training category and word embedding jointly. Doc2Vec is a generalization of Word2Vec model to go beyond word-level to achieve phrase level or sentence level of distributed vector representation. In Doc2Vec, variable length sized text such as documents, paragraphs and sentences are treated as special words and trained with normal words jointly on top of Word2Vec learning framework. Those documents, paragraphs, and sentences act as memory for recording the main topics of co-training words. Correspondingly, in Category2Vec, categories are treated as special words recording the main topic of entity description. The Category2Vec learning framework is illustrated in Fig. 1. Formally, given a sequence of training words $\{w_1, w_2, \ldots w_T\}$ from an entity description, and a sequence of semantic categories $\{c_1, c_2, \ldots c_j\}$ denoting the categorical feature of entity, word vectors and category vectors are trained jointly into a same distributed vector space by maximizing the average log probability:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\le k\le c, k\neq 0} logp(w_t|c_1, c_2, \ldots c_j, w_{t-k}, \ldots, w_{t+k}) \tag{14}$$

where $p(w_t|c_1, c_2, \ldots c_j, w_{t-k}, \ldots, w_{t+k})$ is the hierarchical softmax of the word vectors and category vectors (Le & Mikolov, 2014), while k is the context window size. The category vectors and word vectors are trained using stochastic gradient descent and the gradient is obtained via back propagation (Rumelhart, Hinton, & Williams, 1988). The category vectors contribute to the prediction task of the next word given contextual words sampled from entity description. The contextual words are fixed-length and sampled from a sliding window over all the entity descriptions in a given KG. As shown in Fig. 1, in Category2Vec learning framework, every category and word are mapped to unique vectors and their average or concatenation are used to predict the next word in a context. The word vectors are shared across all the entities as long as the word occurs in the entity description, while the category vectors are shared across those entities having the same category. As training results, we can obtain both word and category vectors, while word vectors are equivalent to Word2Vec model described in previous section.

In trained Category2Vec model, since word and category vectors are in the same shared vector space, they can be directly used to compute word-category similarity based on cosine similarity of vectors. We use two strategies to develop similarity based NED, namely *max* and *average* strategies. With the *max* strategy, we treat categories as special feature words and use the following function to implement NED approach.

$$\hat{e} = \underset{e_i\in E}{argmax} \sum_{w_i\in C, f_j\in F(e_i)}^{K} sim_{category2vec}(w_i, f_j) \tag{15}$$

where $f_j \in F(e_i)$ denotes the categories of entity e_i, and the similarity function $sim_{category2vec}$ is cosine similarity between word and category in the joint vector space learned from Category2Vec. In this strategy, Category2Vec is used as another corpus-based similarity method while it computes word-category similarity. In the *average* strategy, contextual words and entity categories are first mapped to corresponding vectors, and then combined respectively with normalized average. Then, the context and entity similarity is computed based on cosine similarity between averaged context and entity vector.

$$\hat{e} = \underset{e_i\in E}{argmaxsim_{cosine}}(avg(C), avg(F(e_i))) \tag{16}$$

where $avg(C)$ and $avg(F(e_i))$ are the normalized average vectors of contextual words and categories. This strategy is actually simulating the computation of text similarity. We will compare *max* strategy and *average* strategy in our experiment.

Moreover, in the Category2Vec model, since categories are treated as special words for recording the semantic topics of a group of words (concatenation of textual words from list of entity descriptions), compared to latent topics in topic models (Kataria et al., 2011), categories can be viewed as explicit topics representing human defined domain knowledge. Comparing to entity embedding (Fang, Zhang, Wang, Chen, & Li, 2016; Zwicklbauer, Seifert, & Granitzer, 2016) based on entity-entity co-occurrence (Nunes et al., 2013) and entity distribution (Aggarwal et al., 2015; Shen et al., 2012a), category embedding is independent of annotated data and can have more training data by collecting multiple entities which share the same category. In KGs, an entity usually has multiple categories from general to specific describing different aspects of entity. For example, entity *dbr:John_Noble* has categories from general to specific, dbo:Person, yago:Director, dbc:Australian_Film_Actor. In addition, a category normally annotates multiple entities indicating their common categorical feature (e.g. all the director entities have category *yago:Director*). Because of these characteristics of entities and categories in KG, category vectors would be used more frequently since they are shared by multiple entities, while entity vectors consume more storage space but are used less frequently. Also, combining multiple category vectors to represent an entity vector can capture a more complete meaning of an entity because it combines entity's different aspects. In this way, entity vectors constructed from category vectors may be more effective than those entity vectors from entity embedding. In addition, given that categories are usually constructed hierarchically from general to specific into concept taxonomies, more general categories subsume more specific categories. Correspondingly, entities are annotated with categories from general to specific. In consequence, while training Category2Vec model, more general categories appear more frequently, thus they are related to more various words, whereas more specific categories have less collocation with words. Examples of similar categories and words in trained Category2vec model are shown in Table 4, which uses DBpedia abstracts and categories. The category *dbc:Machine_learning* is sub-category of *dbc:Artificial_intelligence*. By showing their top 15 similar categories and words, we illustrate that the more general category is more similar to general categories and words, while

Table 4
Examples of top-15 similar categories and words of Category2Vec model trained in DBpedia.

dbc:Artificial_intelligence				dbc:Machine_learning			
Category	Similarity	Word	Similarity	Category	Similarity	Word	Similarity
dbc:Machine_learning	0.761	application	0.613	dbc:Classification_algorithms	0.774	algorithm	0.561
dbc:Cognitive_science	0.709	process	0.581	dbc:Artificial_intelligence	0.761	logical	0.548
dbc:Information_retrieval	0.658	intelligent	0.575	dbc:Machine_learning_algorithms	0.758	computation	0.545
dbc:Semantic_Web	0.658	analysis	0.565	dbc:Structured_prediction	0.706	analysis	0.540
dbc:Artificial_neural_networks	0.654	methodology	0.563	dbc:Computational_learning_theory	0.705	numerical	0.525
dbc:Data_modeling	0.652	algorithm	0.552	dbc:Learning	0.683	parameter	0.522
dbc:Knowledge_engineering	0.651	logic	0.547	dbc:Artificial_neural_networks	0.667	method	0.519
dbc:Automated_planning_and_scheduling	0.638	knowledge	0.543	dbc:Computational_complexity_theory	0.649	heuristic	0.505
dbc:Information_systems	0.635	simulation	0.543	dbc:Cognitive_science	0.639	predictive	0.498
dbc:Learning	0.631	interaction	0.528	dbc:Learning_methods	0.627	process	0.497
dbc:Semantics	0.629	learn	0.521	dbc:Decision_theory	0.625	calculation	0.495
dbc:Natural_language_processing	0.628	cognition	0.505	dbc:Algorithms_and_data_structures	0.623	unsupervised	0.491
dbc:Decision_theory	0.624	communicate	0.501	dbc:Statistical_natural_language_processing	0.619	knowledge	0.486
dbc:Simulation	0.609	heuristic	0.500	dbc:Data_mining	0.604	learn	0.479
dbc:Model_checkers	0.608	mathematic	0.482	dbc:Decision_trees	0.599	mathematic	0.476

more specific category is more similar to those specific categories and words. The comparison example shows the property of Category2Vec in capturing the generality and specificity of categories in shared vector space.

In summary, Category2Vec model groups larger numbers of general words into general categories and smaller numbers of specific words into specific categories. In other words, general categories can be viewed as a larger word cluster and specific categories correspond to a smaller word cluster which is contained in the larger word cluster. With this property, Category2Vec model can be used to compare various semantic distance for word-word, word-category, and category-category. In this article, we mainly investigate the effectiveness in measuring word-word and word-category similarity for NED.

4. Evaluation

We evaluate the proposed methods with various datasets and answer the research questions through experimental result analysis.

4.1. Datasets and implementations

We collected three NEL datasets which are publicly available, including search engine queries, web question answering queries and tweets. As we mainly focus on evaluating the effectiveness of similarity-based NED, we processed the original datasets according to two criteria: (1) the annotated mentions are ambiguous and have more than one candidates in DBpedia; (2) the mention contexts have at least one common noun. The first criteria is used for testing NED specially, while the second criteria is used to create a fair comparison framework for all the similarity methods since knowledge-based semantic similarity methods mostly work for nouns in WordNet. We describe the details of dataset preparation correspondingly for each datasets as below:

Web Queries dataset (Hasibi, Balog, & Bratsberg, 2015) contains queries derived from *Y-ERD* that offers 2398 entity-annotated queries collecting from the entity recognition and disambiguation challenge (Carmel et al., 2014) and Yahoo Search Query Log to Entities (Hasibi et al., 2015). We only remain those instances that have been annotated with DBpedia entities resulted in 1151 instances. After filtering based on our criteria, we finally got 340 queries for our experiment. Note that since queries are too short, we also include context words from entity mentions (e.g. contextual words music and song are extracted from entity mention “music man song”).

Web Questions dataset (Berant et al., 2013) contains thousands of question answer pairs and each question has been annotated with a Freebase (Bollacker et al., 2008) entity. We have converted those Freebase entities to the corresponding DBpedia entities using a mapping dataset provided by DBpedia (same-as relation). From the original 3778 training dataset, we successfully mapped 2019 entities to DBpedia. After filtering, we finally got 587 questions for our experiment.

Tweets (Cano, Preotiuc-Pietro, Radovanović, Weller, & Dadzie, 2016) dataset is the entity linking dataset of Named Entity rEcognition and Linking Challenge (NEEL) at the #Microposts2016, whose task consists of recognizing named entity mentions and their types from English tweets, and linking them to corresponding DBpedia entries with proper entity disambiguation. We use the training dataset containing 6025 annotated English tweets and after filtering we have got 2284 instance for our experiment.

The details of dataset statistics and example text of each dataset are illustrated in Table 5. We have shown the average number of contextual words in column #Context and average number of candidate entities in column #Candidates. The column #NED-C1 shows the number of instances after filtering with criterion 1, while the column #NED-C2 denotes criterion 2. As expected, the tweet dataset contains more contextual words and ambiguous candidates due to its relatively larger size of text and open domain entities. The web query dataset contains more ambiguous candidates than the question dataset because its entity mentions represent many products such as movies, novels, albums, and those products have different versions. Moreover, in web queries dataset, we have considered the words in entity mentions as contextual words since they contain meaningful common nouns (e.g. song and novel). In case of question dataset, the factual queries focus on asking questions about people, place and organization, thus the entity mentions are mostly proper nouns, while the ambiguity level of mentions is relatively lower than other two datasets.

All the datasets described above contain original text, annotated mentions, and corresponding gold standard DBpedia entities, whereas the candidate entities are missing. In order to generate candidates for each annotated mention, we created an entity name dictionary using DBpedia datasets of English titles[2] and redirects.[3] The DBpedia titles are actually created from Wikipedia page titles and the redirects are extracted from the redirect pages consisting of a redirection hyperlink from an alternative name to the article indicating synonyms or aliases such as acronyms, common

[2] http://wiki.dbpedia.org/Downloads2015-04#titles.

[3] http://wiki.dbpedia.org/Downloads2015-04#redirects.

Table 5
Dataset statistics.

Dataset	#Orignal	#NEL	#NED-C1	#NED-C2	#Context	#Candidates	Example
Web Queries	2398	1151	361	336	2.2	8.4	“the music man songs”
Web Questions	3778	2019	681	607	1.9	6.7	“What movies did John Noble play”
Tweets	6025	6025	2497	2289	3.2	9.6	“Five New Apple Retail Stores Opening Around the World.”

misspellings to name a few. By mapping entity resources in two datasets, we have created an entity name dictionary where various forms of entity names are mapped to a set of DBpedia entities sharing the same lexical names. Then, entity candidates are generated for each dataset by performing exact string matching over annotated entity mentions and entity names in the dictionary. To guarantee the filtering criteria in preparing datasets, those entity mentions having no matching of candidate entities from the entity dictionary, have been removed from the datasets. Thus, all the mentions would have at least two candidate entities. Those entity mentions having only one single matched entity are discarded, since we are focusing on testing the performance of NED. We use English abstracts[4] and categories[5] of DBpedia as NED features for different semantic similarity models. To develop IR and LSA models, we use Gensim[6] to process entity abstracts and index them. For those Natural Language Processing (NLP) tasks of tokenization, part of speech tagging, we use the spaCY,[7] while the lemmatisation is based on NLTK.[8] Moreover, we use Sematch (Zhu & Iglesias, 2017b)[9] tool to compute knowledge-based semantic similarity of words using WordNet, while we use the implementation of Word2Vec from Gensim for training word embedding. Category2Vec is built based on the Doc2Vec implementation of Gensim. By joining entity abstracts and categories, we trained category and word embedding using the CBOW with 300 dimensions.

4.2. Experimental settings and results analysis

Through the evaluation of NED baseline approaches (TF-IDF and LSA), SCSNED, and Category2Vec in the prepared datasets mentioned above, we want to address the following research questions (RQs):

- RQ1: How do different NED approaches compare with different type of texts?
- RQ2: How do different word similarity methods compare with the task of NED?
- RQ3: How do different knowledge-based similarity methods compare with the task of NED?
- RQ4: How do different training corpus affect the performance of word embedding on the task of NED?
- RQ5: How do average and max similarity strategy compare with Category2Vec model in the task of NED?

In order to answer the above research questions, we have implemented all the NED approaches described in Section 3 and evaluated them with three types of datasets. To answer RQ2, we tested SCSNED approach with both corpus-based and knowledge-based word similarity methods. Specifically, all the knowledge-based similarity methods mentioned in Section 2.2 are tested respectively using WordNet. For the corpus-based similarity, to answer the RQ4, we collected three word embeddings which are trained from GoogleNews, Wikipedia and DBpedia abstracts respectively. The original word embedding results of Word2Vec trained in GoogleNews (Mikolov et al., 2013) is used to represent open domain word embedding, while we trained Wikipedia-based word embedding using English Wikipedia with Gensim Word2Vec tool. As Category2Vec is an extension to Word2Vec, we use the word embedding trained in Category2Vec module based on entity abstracts of DBpedia. In addition, we have implemented and evaluated both average and max similarity strategy of NED using Category2Vec model. We use the standard accuracy, precision, recall and *F*-measure as metrics for NED (Navigli, 2009), and the evaluation results are shown in Table 6. We present the conclusion to answer each research question respectively according to the evaluation results reported in the table.

RQ1. Firstly, the SCSNED and Category2Vec approaches have better performance than baseline approaches in all types of datasets, which shows that the fine-grained meaning comparison is more effective than coarse-grained meaning comparison in task of NED. Secondly, as expected, the LSA has been shown better performance than the basic TF-IDF model, thus we draw conclusion that the dimension reduction is effective in solving vocabulary mismatch problem. Thirdly, word-category similarity methods is relatively better than word-word similarity method in shorter texts such as web queries and web questions, while word-word similarity methods are better in relatively longer texts (e.g. tweets). We think this is because the category vectors have more specific meaning than common words in discriminating the entities according to the contextual words. For example, category *dbc:Machine_learning* and category *dbc:Classification_algorithms* are more specific than words *machine, learning, classification, algorithm*. Thus, when the context is limited, more specific meaning would play more important role in deciding the correct meaning of the entity. Finally, as we have used a voting strategy (we used top-10 word similarity) in word-word similarity-based NED, with the relatively more contextual words, word-word similarity methods have better performance since entities have more word features than category features.

RQ2. Knowledge-based similarity methods transform the structural knowledge contained in WordNet into similarity scores, while corpus-based Word2Vec transforms statistical knowledge into similarity scores. Through those similarity scores, external resources such as WordNet, and textual corpora such as Wikipedia, GoogleNews are employed to help NED. With WordNet, since the semantic relations between concepts are fixed in the ontology, the effectiveness of knowledge transformation relies on the effectiveness of similarity methods. In comparison, the effectiveness of Word2Vec model depends on the proper textual corpus for training. The experimental results have shown that corpus-based and knowledge-based similarity methods have similar performance in all types of datasets, since we have only considered general domains (e.g. web queries, web questions and tweets). Comparing the best results obtained by the two types of similarity methods, knowledge-based similarity methods have better results in web queries and tweets, while corpus-based similarity methods have better results in web questions. We think that corpus-based similarity methods capture better relatedness in questions (e.g. movie and actor, shown in Table 2), while knowledge-based similarity methods represent more information in same domain (movie, novel, or place). Thus,

[4] http://wiki.dbpedia.org/Downloads2015-04#extended-abstracts.
[5] http://wiki.dbpedia.org/Downloads2015-04#articles-categories.
[6] https://radimrehurek.com/gensim/.
[7] https://spacy.io/.
[8] https://www.nltk.org.
[9] https://github.com/gsi-upm/sematch.

Table 6
NED in various sizes of texts using different similarity methods.

Method	Web queries				Web questions				Tweets			
	Accuracy	Precision	Recall	*F*-Measure	Accuracy	Precision	Recall	*F*-Measure	Accuracy	Precision	Recall	*F*-Measure
TF-IDF	0.421	0.582	0.421	0.459	0.496	0.523	0.496	0.503	0.338	0.671	0.338	0.397
LSA	0.459	0.626	0.459	0.500	0.579	0.607	0.579	0.588	0.399	0.698	0.399	0.470
Word2Vec DBpedia	0.453	0.604	0.453	0.489	0.733	0.743	0.733	0.735	0.440	0.683	0.440	0.493
Word2Vec Wikipedia	0.524	0.676	0.524	0.561	0.753	0.761	0.753	0.755	0.468	0.709	0.468	0.523
Word2Vec GoogleNews	0.500	0.643	0.500	0.539	0.698	0.710	0.698	0.702	0.490	0.701	0.490	0.542
WordNet Path	0.529	0.616	0.529	0.555	0.734	0.740	0.734	0.736	0.474	0.668	0.474	0.524
WordNet Wu & Palmer	0.494	0.592	0.494	0.519	0.700	0.706	0.700	0.702	0.447	0.634	0.447	0.491
WordNet Resnik	0.508	0.611	0.508	0.541	0.716	0.722	0.716	0.717	0.451	0.656	0.451	0.501
WordNet Lin	0.535	0.617	0.535	0.560	0.721	0.725	0.721	0.721	0.480	0.662	0.480	0.527
WordNet Jiang & Conrad	0.538	0.604	0.538	0.562	0.717	0.723	0.717	0.719	0.500	0.699	0.500	0.548
WordNet WPath	0.532	0.616	0.532	0.556	0.733	0.743	0.733	0.735	0.462	0.663	0.462	0.512
Category2Vec Average	0.529	0.738	0.529	0.581	0.600	0.626	0.600	0.606	0.360	0.687	0.360	0.430
Category2Vec Max	0.591	0.717	0.591	0.630	0.765	0.766	0.765	0.765	0.483	0.682	0.483	0.529

it is better to combine both types of similarity methods in order to contain both relatedness and domain knowledge. Moreover, the knowledge-based similarity methods are based on a specific domain ontology which has limited their application to the scenarios having a well constructed domain ontology. As ontologies are difficult to develop and maintain, corpus-based similarity methods can play important role as the complement of knowledge-based similarity methods. Furthermore, because of simple computational model, corpus-based similarity methods are easy to apply for covering more updated vocabularies.

RQ3. Previous works of semantic similarity evaluations are based on word similarity dataset containing word pairs which are assigned with similarity scale by human subjects (Zhu & Iglesias, 2017a). Such evaluation relies on human evaluation over word pairs which may not have the same performance in real applications (Budanitsky & Hirst, 2006). In the task of NED, we are able to compare the effectiveness of different knowledge-based similarity methods in real world application and datasets. In general, from the experimental results, we have found that Resnik (1995) and Wu and Palmer (1994) methods have relatively lower effectiveness than other four methods in task of NED. Although the other four similarity methods perform differently in different datasets, the Jiang and Conrath (1997) method has relatively better performance in queries and tweets, while the Path (Rada et al., 1989) method performs better in web questions. Since WPath (Zhu & Iglesias, 2017a) is a tradeoff between the Path (Rada et al., 1989) and pure IC-based methods, it has been shown with consistent performance in all the datasets, which is better than negative cases of IC-based methods and the Path (Rada et al., 1989), but is lower than their positive cases.

RQ4. Word embedding methods are previously evaluated on word level applications such as word similarity, word analogy, part of speech tagging to name a few. Through the NED evaluation, we are able to show the performance of embedding methods in terms of the different corpus for the real application. In general, more training data, the better performance of corpus-based similarity methods. As expected, to train Word2Vec model, more training data (e.g. Wikipedia and GoogleNews) has better performance than less training data (e.g. DBpedia abstract). Since the domain information also influences the training of word embedding, the word embedding trained from DBpedia abstracts and Wikipedia have better performance than GoogleNews in web queries and web questions. In a more general domain, tweets, the general corpus GoogleNews has performed best. Through such comparison, we demonstrate that the different characteristics of training corpus would affect the application of word embeddings in specific domain application such as NED. Furthermore, as Category2Vec is literally an extension of Word2Vec model, it can obtain both category embedding and word embedding from a unified training model. Since both word vectors and category vector are embedded in a shared vector space, Category2Vec would have more applications than word embedding alone. Note that category-category relation can be used to infer entity-entity relation. In conclusion, we have shown that Category2Vec is suitable for training embedding in KGs where both semantic category and textual descriptions are available for entities.

RQ5. We have provided average and maximum strategies in applying the Category2Vec model for NED and evaluated them in all the datasets respectively. The experimental results have shown that the maximum strategy is better than average one in case of NED. We analyze the results empirically by comparing two strategies. For the average strategy, the contextual words and entity categories are generalized via averaging their respective vectors, in order to test the performance of vector combined meaning. The normalized sum average of contextual words and entity categories seem to be too generalized for representing the specific meaning of the context and entity, while the maximum strategy of selecting most similar word-category pairs between context and entity has a better performance. Each entity category represents a specific aspect of the entity, while each contextual word also represents a specific aspect of the context meaning. Finding the most similar pair of contextual word and entity category is actually aligning the context meaning with entity meaning in terms of specific aspects. In this semantic alignment process, individual word or category vectors help to identify the distinctive similarity between context and entity. In consequence, the max strategy is more effective than simple average strategy in representing distinctive word and category meanings. We think that the average strategy has lower performance in representing discriminative features, whereas it may be more suitable in the case that the generalized meaning representation is needed such as detecting the word analogy or representing generative features.

To summarize, we have answered the research questions through the experiment in the task of NED of web queries, web questions and tweets. We have demonstrated: (1) fine-grained word level meaning comparison is better than the baseline of text level meaning comparison (e.g. TF-IDF and LSA) in case of short texts; (2) both knowledge-based and corpus-based similarity methods are effective for NED while it is better to combine both; (3) knowledge-based similarity methods, Path (Rada et al., 1989), Jiang and Conrath (1997) and WPath (Zhu & Iglesias, 2017a) are generally better than others in case of NED; (4) corpus-based similarity methods depend on proper selection of domain corpus, while Category2Vec is more effective in generating and representing both word and category vectors in a shared vector space with a uniformed embedding model; (5) the Category2Vec model is ef-

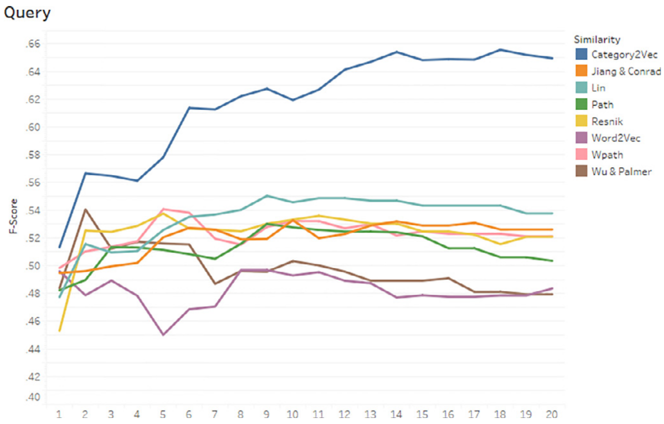

Fig. 2. *F*-score comparison by changing K in query dataset.

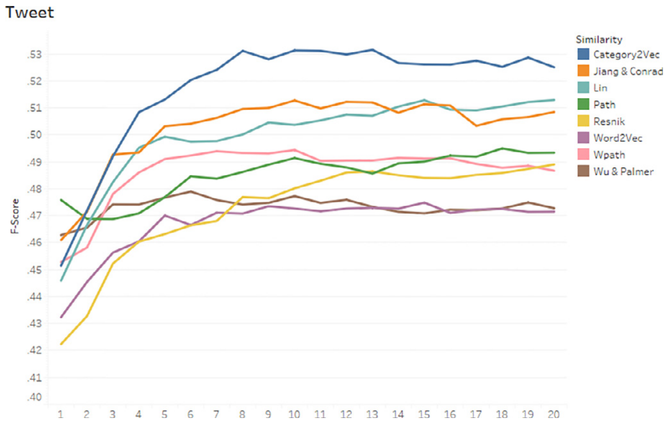

Fig. 4. *F*-score comparison by changing K in tweet dataset.

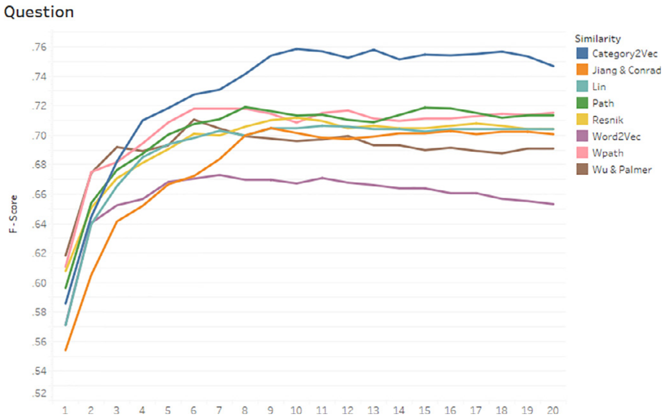

Fig. 3. *F*-score comparison by changing K in question dataset.

fective for NED, while the maximum strategy is better than average strategy in case of similarity-based NED.

4.3. Comparing K settings for similarity methods

Since SCSNED and Category2Vec methods use top *K* similarity score, we evaluate disambiguation performance of using different similarity method when *K* is changing from 1 to 20. We use entity categories as entity features and run experiments in query, question, and tweet datasets. The similarity methods used in experiments include Category2Vec, Word2Vec (Mikolov et al., 2013), Path (Rada et al., 1989), Wu and Palmer (1994), Resnik (1995), Jiang and Conrath (1997),Lin (1998), and WPath (Zhu & Iglesias, 2017a). From Fig. 2, Fig. 3 and Fig. 4, we can see that *F*-score is generally increasing as *K* increases. With larger *K* values, *F*-score is tending to become smaller and keep steady. Moreover, Category2Vec performs better than other similarity methods in terms of using category features, showing its advantage in computing word and category similarity.

4.4. Comparing to the state of the art

Having evaluated different similarity methods of SCSNED and Category2Vec, we provide an evaluation of proposed NED approaches to the state of the art NED approaches with all the tweet dataset (6025 instances) in NEEL 2016 and RSS 500 (Röder, Usbeck, Hellmann, Gerber, & Both, 2014), because their evaluation results are publicly available. NEEL 2016 is used for comparing short text case while RSS 500 is used for comparing document case. We follow the *strong link match* evaluation (Cano et al., 2016) which specially evaluates the disambiguation effectiveness, where the mentions and candidates are given. The entity disambiguation approaches are applied when an entity mention has multiple candidate entities from dictionary matching such as John Noble has multiple candidates shown in Table 1. NIL addresses problem that an entity mention has no matching in the given KG. We simply address NIL case based on candidate generation where candidates are generated based on exact string matching to entity dictionary. If the entity dictionary matches, we perform disambiguation. If there is no match, we simply tag NIL. We do not specially address the NIL problem because it is related to many factors. To have an acceptable comparison, we use simple matching approach. Note that in the *strong link match* evaluation, a system needs to map the given entity mention to DBpedia entity candidates and select the most suitable entity, thus the final performance is also influenced by the entity name dictionary and name matching apart from the disambiguation algorithm. We use the whole name dictionary constructed from DBpedia title dataset which is mentioned previously. In the evaluation described in the previous section, we have avoided such influence, while we include this factor in this section in order to compare to the existing approaches. We use the GERBIL (Usbeck et al., 2015) framework to retrieve the evaluation results of the same tweet dataset and RSS 500 for those state of the art NEL systems who have separate NED module available. The *D2KB* model in GERBIL framework is chosen because it is the corresponding evaluation mode that is equivalent to *strong link match*. From NEEL 2016 report (Cano et al., 2016), we get the results of challenge's baseline system ADEL (Plu, Rizzo, & Troncy, 2016) whose disambiguation approach is based on string similarity and graph-based algorithm PageRank (Hachey et al., 2011). We also get the results of the system KEA (Waitelonis & Sack, 2016) (best reported system in the challenge), whose disambiguation is based on confidence scoring considering string similarity, weighting graph distance, connected component analysis, entity centrality and density.

From GERBIL, we have retrieved the evaluation results of AGDISTIS (Usbeck et al., 2014), AIDA (Hoffart et al., 2011), Babelfy (Moro et al., 2014), PBOH (Ganea et al., 2016), WAT (Piccinno & Ferragina, 2014), and DoSeR (Zwicklbauer et al., 2016), which have separate NED component. AGDISTIS (Usbeck et al., 2014) is based on string similarity and graph-based HITS algorithm, while AIDA (Hoffart et al., 2011) combines entity prior probability, keyphrase-based context similarity and entity coherence. Babelfy (Moro et al., 2014) models entities in a network through its "semantic signature" based on graph random walk algorithm and identifies entity through iterative process in the subgraph. PBOH (Ganea et al., 2016) is a recent pure NED approach that develops probabilistic graphical model using pairwise Markov Random Fields for disam-

Table 7
Performance comparison of state-of-the-art systems and the proposed methods.

System	VoteCombine	Category2Vec	LSA	Word2Vec	WordNet	AGDISTIS	AIDA	Babelfy	PBOH	WAT	DoSeR	KEA	ADEL
Tweet	0.595	0.579	0.557	0.582	0.585	0.561	0.489	0.428	0.721	0.587	0.607	0.501	0.536
RSS-500	0.583	0.591	0.572	0.574	0.577	0.61	0.43	0.45	0.721	0.44	0.607	0.501	NA

biguation based on statistics from English Wikipeda corpus considering anchor text. WAT (Piccinno & Ferragina, 2014) is a redesigned system of TagMe (Ferragina & Scaiella, 2012) and includes graph-based algorithm for ranking entities in entity graph based on entity relatedness, and vote-based algorithm for local disambiguation. DoSeR (Zwicklbauer et al., 2016) is another recent entity disambiguation framework, which combines entity prior probability, entity relatedness and PageRank based disambiguation algorithm. Its entity relatedness is measured based on cosine similarity of entity embedding vectors which is trained from generated entity sequence corpus using Word2Vec (Mikolov et al., 2013). In comparison, Category2Vec trains category and word embedding. We have discussed and compared entity embedding and category embedding in Section 3.3. These state of art systems cover most of disambiguation approaches described in Section 2.1.2.

We include LSA, SCSNED and Category2Vec (max strategy) to compare with existing systems. For SCSNED, we use the two best performing word similarity models in tweet data, which are Word2Vec based on GoogleNews corpus and Jiang and Conrath (1997) similarity method based on WordNet (referred as Word2Vec and WordNet respectively for convenience). Furthermore, we use a simple voting-based ensemble approach, called VoteCombine to combine LSA, Word2Vec, WordNet, and Category2Vec. Since each disambiguation approach returns one single best entity, the VoteCombine chooses the majority one from the four approaches. If the majority vote fails, such as each approach returns four different entities or two groups of approaches return two different entities, we use the one from WordNet based SCSNED approach. The F scores of all the NED systems in tweet dataset and RSS 500 dataset are shown in Table 7. The evaluation results show that all the proposed NED approaches outperforms NEEL challenge baseline and are competitive to the state of the art approaches, while the proposed NED approaches are also applicable to longer text case. The VoteCombine has better performance than most existing systems but PBOH and DoSeR, because those two systems employ more features such as entity prominence, PageRank, probabilistic coherence, and they are trained as unified disambiguation system from entity-annotated dataset. The proposed word-word and word-category similarity methods can be effective features to complement such systems. We aim to provide the proposed approaches as individual module for disambiguation specially so that other approaches using different mention detection or disambiguation methods can use the proposed semantic model as well. Moreover, since SCSNED and Category2Vec only rely on the information contained in KGs themselves, and do not rely on additional entity-annotated dataset which are constructed based on annotating entities in a given KG to textual corpus, they can be directly applied to other KGs that have no entity-annotated dataset. In summary, by comparing to the current state of the art NED system, we have shown that the proposed similarity-based NED approaches are effective and useful. Especially, the studied word similarity methods and Category2Vec model are effective additional features to be complement of the current systems.

5. Conclusions and future works

In this paper, we have exploited different semantic similarity methods based on various semantic resources, including self-contained KG features (e.g. entity abstract and category), common sense knowledge from ontology (e.g. WordNet) and textual corpora (e.g. GoogleNews and Wikipedia). We proposed a novel NED approach based on word similarity and evaluated both knowledge-based and corpus-based semantic similarity methods in the task of NED. We have demonstrated and identified effectiveness of different semantic similarity methods in a comparative experiment of evaluating unsupervised similarity-based NED with real world datasets of web queries, web questions and tweets. Moreover, we proposed Category2Vec model to learn vector representation of words and categories jointly in the same shared vector space only dependent on a uniformed embedding model and knowledge of KG (e.g. entity abstracts and categories), without relying on labeled dataset. All the similarity methods and models presented in this paper can also be used in other KG-based applications which require to explore similarity between word, category and entities. Our main experimental results have shown that semantic similarity methods are more effective than text similarity methods when the contextual information is scarce. The proposed SCSNED and Category2Vec methods are competitive to the state of the art NED approaches, while the semantic similarity features are shown to be effective and can be used as the complement to the existing approaches.

As this paper mainly exploits semantic similarity methods for NED in order to identify their effectiveness, it would be interesting future work to evaluate the performance of similarity methods when they are combined with other NED methods and features, and integrated into a complete NEL system or other KG-based applications. Especially, in the future work, we will evaluate the performance of using category and word vectors for measuring entity-entity relatedness by comparing to other common entity-entity relatedness methods. Moreover, since DBpedia is used as an illustrative example in this work, another future work might be the application of similarity methods presented in this work to the other domain KGs given that the described similarity methods are only dependent on self-contained KG features and common sense knowledge.

Acknowledgements

This work is supported by the Spanish Ministry of Economy and Competitiveness under the R&D projects SEMOLA (TEC2015-68284-R) and EmoSpaces (RTC-2016-5053-7), by the Regional Government of Madrid through the project MOSI-AGIL-CM (grant P2013/ICE-3019, co-funded by EU Structural Funds FSE and FEDER), and by the European Union through the project MixedEmotions (Grant Agreement no: 141111).

References

Aggarwal, N., Asooja, K., Ziad, H., & Buitelaar, P. (2015). Who are the American vegans related to brad pitt?: Exploring related entities. In *Proceedings of the 24th international conference on world wide web* (pp. 151–154). ACM.

Agirre, E., & Soroa, A. (2009). Personalizing pagerank for word sense disambiguation. In *Proceedings of the 12th conference of the European chapter of the association for computational linguistics* (pp. 33–41). Association for Computational Linguistics.

Alhelbawy, A., & Gaizauskas, R. (2014). Collective named entity disambiguation using graph ranking and clique partitioning approaches. In *Proceedings of COLING 2014, the 25th international conference on computational linguistics: Technical papers* (pp. 1544–1555).

Baeza-Yates, R., Ribeiro-Neto, B., et al. (1999). *Modern information retrieval*: 463. ACM press New York.

Baroni, M., Dinu, G., & Kruszewski, G. (2014). Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In *Acl (1)* (pp. 238–247).

Bekkerman, R., & Gavish, M. (2011). High-precision phrase-based document classification on a modern scale. In *Proceedings of the 17th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 231–239). ACM.

Berant, J., Chou, A., Frostig, R., & Liang, P. (2013). Semantic parsing on freebase from question-answer pairs. In *EMNLP* (pp. 1533–1544).

Bizer, C., Heath, T., & Berners-Lee, T. (2009). Linked data-the story so far. *Semantic Services, Interoperability and Web Applications: Emerging Concepts*, 205–227.

Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., et al. (2009). Dbpedia - a crystallization point for the web of data. *Web Semantics: Science, Services and Agents on the World Wide Web, 7*(3), 154–165. The Web of Data. doi:10.1016/j.websem.2009.07.002.

Blanco, R., Ottaviano, G., & Meij, E. (2015). Fast and space-efficient entity linking for queries. In *Proceedings of the eighth ACM international conference on web search and data mining, WSDM '15* (pp. 179–188). New York, NY, USA: ACM. doi:10.1145/2684822.2685317.

Bollacker, K., Evans, C., Paritosh, P., Sturge, T., & Taylor, J. (2008). Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on management of data, SIGMOD '08* (pp. 1247–1250). New York, NY, USA: ACM. doi:10.1145/1376616.1376746.

Budanitsky, A., & Hirst, G. (2001). Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures. *Workshop on wordnet and other lexical resources*: 2. 2–2.

Budanitsky, A., & Hirst, G. (2006). Evaluating wordnet-based measures of lexical semantic relatedness. *Computational Linguistics, 32*(1), 13–47.

Bunescu, R. C., & Pasca, M. (2006). Using encyclopedic knowledge for named entity disambiguation. In *EACL: 6* (pp. 9–16).

Cano, A. E., Preotiuc-Pietro, D., Radovanović, D., Weller, K., & Dadzie, A.-S. (2016). #microposts2016: 6th workshop on making sense of microposts: Big things come in small packages. In *Proceedings of the 25th international conference companion on world wide web* (pp. 1041–1042). doi:10.1145/2872518.2893528.

Carmel, D., Chang, M.-W., Gabrilovich, E., Hsu, B.-J. P., & Wang, K. (2014). Erd'14: Entity recognition and disambiguation challenge. In *ACM SIGIR forum: 48* (pp. 63–77). ACM.

Ceccarelli, D., Lucchese, C., Orlando, S., Perego, R., & Trani, S. (2013). Learning relatedness measures for entity linking. In *Proceedings of the 22nd ACM international conference on information & knowledge management, CIKM '13* (pp. 139–148). New York, NY, USA: ACM. doi:10.1145/2505515.2505711.

Church, K. W., & Hanks, P. (1990). Word association norms, mutual information, and lexicography. *Computational Linguistics, 16*(1), 22–29.

Cilibrasi, R. L., & Vitanyi, P. M. B. (2007). The google similarity distance. *IEEE Transactions on Knowledge and Data Engineering, 19*(3), 370–383. doi:10.1109/TKDE.2007.48.

Cornolti, M., Ferragina, P., & Ciaramita, M. (2013). A framework for benchmarking entity-annotation systems. In *Proceedings of the 22nd international conference on world wide web, WWW '13* (pp. 249–260). New York, NY, USA: ACM. doi:10.1145/2488388.2488411.

Cucerzan, S. (2007). Large-scale named entity disambiguation based on wikipedia data. In *Emnlp-conll: 7* (pp. 708–716).

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science, 41*(6), 391.

Derczynski, L., Maynard, D., Rizzo, G., van Erp, M., Gorrell, G., Troncy, R., et al. (2015). Analysis of named entity recognition and linking for tweets. *Information Processing & Management, 51*(2), 32–49.

Dredze, M., McNamee, P., Rao, D., Gerber, A., & Finin, T. (2010). Entity disambiguation for knowledge base population. In *Proceedings of the 23rd international conference on computational linguistics, COLING '10* (pp. 277–285). Stroudsburg, PA, USA: Association for Computational Linguistics.

Fang, W., Zhang, J., Wang, D., Chen, Z., & Li, M. (2016). Entity disambiguation by knowledge and text jointly embedding. *CoNLL 2016*, 260.

Fernandez Garcia, N., Arias Fisteus, J., & Sanchez Fernandez, L. (2014). Comparative evaluation of link-based approaches for candidate ranking in link-to-wikipedia systems. *Journal of Artificial Intelligence Research, 49*, 733–773.

Ferragina, P., & Scaiella, U. (2010). Tagme: On-the-fly annotation of short text fragments (by wikipedia entities). In *Proceedings of the 19th acm international conference on information and knowledge management, CIKM '10* (pp. 1625–1628). New York, NY, USA: ACM. doi:10.1145/1871437.1871689.

Ferragina, P., & Scaiella, U. (2012). Fast and accurate annotation of short texts with wikipedia pages. *Software, IEEE, 29*(1), 70–75. doi:10.1109/MS.2011.122.

Finkel, J. R., Grenager, T., & Manning, C. (2005). Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of the 43rd annual meeting on association for computational linguistics, ACL '05* (pp. 363–370). Stroudsburg, PA, USA: Association for Computational Linguistics. doi:10.3115/1219840.1219885.

Francis, W. N., & Kucera, H. (1979). *Brown corpus manual*. Brown University.

Francis-Landau, M., Durrett, G., & Klein, D. (2016). Capturing semantic similarity for entity linking with convolutional neural networks. In *Proceedings of naacl-hlt* (pp. 1256–1261).

Ganea, O.-E., Ganea, M., Lucchi, A., Eickhoff, C., & Hofmann, T. (2016). Probabilistic bag-of-hyperlinks model for entity linking. In *Proceedings of the 25th international conference on world wide web* (pp. 927–938). doi:10.1145/2872427.2882988.

Gattani, A., Lamba, D. S., Garera, N., Tiwari, M., Chai, X., Das, S., et al. (2013). Entity extraction, linking, classification, and tagging for social media: A wikipedia-based approach. *Proceedings of the VLDB Endowment, 6*(11), 1126–1137. doi:10.14778/2536222.2536237.

Gligorov, R., ten Kate, W., Aleksovski, Z., & van Harmelen, F. (2007). Using google distance to weight approximate ontology matches. In *Proceedings of the 16th international conference on world wide web* (pp. 767–776). New York, NY, USA: ACM. doi:10.1145/1242572.1242676.

Guo, J., Xu, G., Cheng, X., & Li, H. (2009). Named entity recognition in query. In *SIGIR '09* (pp. 267–274). New York, NY, USA: ACM. doi:10.1145/1571941.1571989.

Guo, S., Chang, M.-W., & Kıcıman, E. (2013). To link or not to link? a study on end-to-end tweet entity linking. *Proceedings of NAACL-HLT*, 1020–1030.

Guo, Y., Che, W., Liu, T., & Li, S. (2011). A graph-based method for entity linking. In *IJCNLP: 1010* (p. 1018).

Gutiérrez, Y., Vázquez, S., & Montoyo, A. (2012). A graph-based approach to wsd using relevant semantic trees and n-cliques model. In *International conference on intelligent text processing and computational linguistics* (pp. 225–237). Springer.

Gutiérrez, Y., Vázquez, S., & Montoyo, A. (2017). Spreading semantic information by word sense disambiguation. *Knowledge-Based Systems*.

Habib, M. B., & van Keulen, M. (2015). Need4tweet: A twitterbot for tweets named entity extraction and disambiguation. *ACL-IJCNLP 2015*, 31.

Hachey, B., Radford, W., & Curran, J. R. (2011). Graph-based named entity linking with wikipedia. In *Web information system engineering–wise 2011* (pp. 213–226). Springer.

Hachey, B., Radford, W., Nothman, J., Honnibal, M., & Curran, J. R. (2013). Evaluating entity linking with wikipedia. *Artificial Intelligence, 194*, 130–150. Artificial Intelligence, Wikipedia and Semi-Structured Resources. doi:10.1016/j.artint.2012.04.005.

Hagen, M., Potthast, M., Stein, B., & Bräutigam, C. (2011). Query segmentation revisited. In *Proceedings of the 20th international conference on world wide web* (pp. 97–106). New York, NY, USA: ACM. doi:10.1145/1963405.1963423.

Han, X., & Sun, L. (2011). A generative entity-mention model for linking entities with knowledge base. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1* (pp. 945–954). Association for Computational Linguistics.

Han, X., Sun, L., & Zhao, J. (2011). *Collective entity linking in web text: A graph-based method, SIGIR '11* (pp. 765–774). New York, NY, USA: ACM. doi:10.1145/2009916.2010019.

Han, X., & Zhao, J. (2009). Named entity disambiguation by leveraging wikipedia semantic knowledge. In *Proceedings of the 18th ACM conference on information and knowledge management, CIKM '09* (pp. 215–224). New York, NY, USA: ACM. doi:10.1145/1645953.1645983.

Hasibi, F., Balog, K., & Bratsberg, S. E. (2015). Entity linking in queries: Tasks and evaluation. In *Proceedings of the 2015 international conference on the theory of information retrieval* (pp. 171–180). ACM.

Hasibi, F., Balog, K., & Bratsberg, S. E. (2016). Exploiting entity linking in queries for entity retrieval. In *Proceedings of the 2016 ACM on international conference on the theory of information retrieval* (pp. 209–218). ACM.

He, Z., Liu, S., Li, M., Zhou, M., Zhang, L., & Wang, H. (2013). Learning entity representation for entity disambiguation. In *ACL (2)* (pp. 30–34).

Hinton, G. E., Osindero, S., & Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation, 18*(7), 1527–1554.

Hoffart, J., Seufert, S., Nguyen, D. B., Theobald, M., & Weikum, G. (2012). Kore: Keyphrase overlap relatedness for entity disambiguation. In *Proceedings of the 21st ACM international conference on information and knowledge management* (pp. 545–554). ACM.

Hoffart, J., Suchanek, F. M., Berberich, K., & Weikum, G. (2013). Yago2: A spatially and temporally enhanced knowledge base from wikipedia. *Artificial Intelligence, 194*, 28–61. Artificial Intelligence, Wikipedia and Semi-Structured Resources. doi:10.1016/j.artint.2012.06.001.

Hoffart, J., Yosef, M. A., Bordino, I., Fürstenau, H., Pinkal, M., Spaniol, M., et al. (2011). Robust disambiguation of named entities in text. In *Proceedings of the conference on empirical methods in natural language processing, EMNLP '11* (pp. 782–792). Stroudsburg, PA, USA: Association for Computational Linguistics.

Houlsby, N., & Ciaramita, M. (2014). Advances in information retrieval. In *36th European conference on IR research, ECIR 2014, Amsterdam, the Netherlands, April 13–16, 2014. proceedings* (pp. 335–346)). Cham: Springer International Publishing. doi:10.1007/978-3-319-06028-6_28.

Hulpuş, I., Prangnawarat, N., & Hayes, C. (2015). Path-based semantic relatedness on linked data and its use to word and entity disambiguation. In *The semantic web-iswc 2015* (pp. 442–457). Springer.

Ji, H., & Grishman, R. (2011). Knowledge base population: Successful approaches and challenges. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1* (pp. 1148–1158). Association for Computational Linguistics.

Jiang, J. J., & Conrath, D. W. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. *Computational Linguistics, cmp-lg/970*(Rocling X), 15.

Kataria, S. S., Kumar, K. S., Rastogi, R. R., Sen, P., & Sengamedu, S. H. (2011). Entity disambiguation with hierarchical topic models. In *Proceedings of the 17th ACM SIGKDD international conference on knowledge discovery and data mining, KDD '11* (pp. 1037–1045). New York, NY, USA: ACM. doi:10.1145/2020408.2020574.

Kulkarni, S., Singh, A., Ramakrishnan, G., & Chakrabarti, S. (2009). Collective annotation of wikipedia entities in web text. In *Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining, KDD '09* (pp. 457–466). New York, NY, USA: ACM. doi:10.1145/1557019.1557073.

Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents. In T. Jebara, & E. P. Xing (Eds.), *Proceedings of the 31st international conference on machine learning (ICML-14)* (pp. 1188–1196). JMLR Workshop and Conference Proceedings.

Lesk, M. (1986). Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proceedings of the 5th annual international conference on systems documentation* (pp. 24–26). ACM.

Levy, O., Goldberg, Y., & Dagan, I. (2015). Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics, 3*, 211–225.

Lin, D. (1998). An information-theoretic definition of similarity. In *Proceedings of the fifteenth international conference on machine learning, ICML '98* (pp. 296–304). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Liu, X., Li, Y., Wu, H., Zhou, M., Wei, F., & Lu, Y. (2013). Entity linking for tweets. In *ACL (1)* (pp. 1304–1311).

Medelyan, O., Witten, I. H., & Milne, D. (2008). Topic indexing with wikipedia. In *Proceedings of the AAAI wikiai workshop: 1* (pp. 19–24).

Meij, E., Bron, M., Hollink, L., Huurnink, B., & de Rijke, M. (2011). Mapping queries to the linking open data cloud: A case study using {DBpedia}. *Web Semantics: Science, Services and Agents on the World Wide Web, 9*(4), 418–433. {JWS} special issue on Semantic Search. doi:10.1016/j.websem.2011.04.001.

Meij, E., Weerkamp, W., & de Rijke, M. (2012). Adding semantics to microblog posts. In *Proceedings of the fifth ACM international conference on web search and data mining* (pp. 563–572). ACM.

Mendes, P. N., Jakob, M., García-Silva, A., & Bizer, C. (2011). Dbpedia spotlight: Shedding light on the web of documents. In *Proceedings of the 7th international conference on semantic systems, I-Semantics '11* (pp. 1–8). New York, NY, USA: ACM. doi:10.1145/2063518.2063519.

Meymandpour, R., & Davis, J. G. (2016). A semantic similarity measure for linked data: An information content-based approach. *Knowledge-Based Systems, 109*, 276–293.

Mihalcea, R., & Csomai, A. (2007). Wikify!: Linking documents to encyclopedic knowledge. In *Proceedings of the sixteenth ACM conference on conference on information and knowledge management, CIKM '07* (pp. 233–242). New York, NY, USA: ACM. doi:10.1145/1321440.1321475.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv:1301.3781.

Miller, G. A. (1995). Wordnet: A lexical database for english. *Communications of the ACM, 38*(11), 39–41.

Milne, D., & Witten, I. H. (2008). Learning to link with wikipedia. In *Proceedings of the 17th ACM conference on information and knowledge management, CIKM '08* (pp. 509–518). New York, NY, USA: ACM. doi:10.1145/1458082.1458150.

Milne, D., & Witten, I. H. (2013). An open-source toolkit for mining wikipedia. *Artificial Intelligence, 194*, 222–239. Artificial Intelligence, Wikipedia and Semi-Structured Resources. doi:10.1016/j.artint.2012.06.007.

Moro, A., Raganato, A., & Navigli, R. (2014). Entity linking meets word sense disambiguation: A unified approach. *Transactions of the Association for Computational Linguistics, 2*, 231–244.

Nadeau, D., & Sekine, S. (2007). A survey of named entity recognition and classification. *Lingvisticae Investigationes, 30*(1), 3–26.

Navigli, R. (2009). Word sense disambiguation: A survey. *ACM Computing Surveys, 41*(2), 10:1–10:69. doi:10.1145/1459352.1459355.

Nunes, B. P., Dietze, S., Casanova, M. A., Kawase, R., Fetahu, B., & Nejdl, W. (2013). Combining a co-occurrence-based and a semantic measure for entity linking. In *The semantic web: Semantics and big data* (pp. 548–562). Springer.

Passant, A. (2010). Measuring semantic distance on linking data and using it for resources recommendations. In *AAAI spring symposium: Linked data meets artificial intelligence: 77* (p. 123).

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the empiricial methods in natural language processing (EMNLP 2014): 12* (pp. 1532–1543).

Piccinno, F., & Ferragina, P. (2014). From tagme to wat: a new entity annotator. In *Proceedings of the first international workshop on entity recognition & disambiguation* (pp. 55–62). ACM.

Pilz, A., & Paaß, G. (2011). From names to entities using thematic context distance. In *Proceedings of the 20th ACM international conference on information and knowledge management, CIKM '11* (pp. 857–866). New York, NY, USA: ACM. doi:10.1145/2063576.2063700.

Pirró, G. (2012). Reword: Semantic relatedness in the web of data. *AAAI*.

Plu, J., Rizzo, G., & Troncy, R. (2016). Enhancing entity linking by combining ner models. In *Semantic web evaluation challenge* (pp. 17–32). Springer.

Pound, J., Hudek, A. K., Ilyas, I. F., & Weddell, G. (2012). Interpreting keyword queries over web knowledge bases. In *Proceedings of the 21st ACM international conference on information and knowledge management, CIKM '12* (pp. 305–314). New York, NY, USA: ACM. doi:10.1145/2396761.2396803.

Pu, K. Q., & Yu, X. (2008). Keyword query cleaning. *Proceedings of the VLDB Endowment, 1*(1), 909–920. doi:10.14778/1453856.1453955.

Rada, R., Mili, H., Bicknell, E., & Blettner, M. (1989). Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man, and Cybernetics, 19*(1), 17–30. doi:10.1109/21.24528.

Ratinov, L., Roth, D., Downey, D., & Anderson, M. (2011). Local and global algorithms for disambiguation to wikipedia. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies - volume 1, HLT '11* (pp. 1375–1384). Stroudsburg, PA, USA: Association for Computational Linguistics.

Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th international joint conference on artificial intelligence - volume 1, IJCAI'95* (pp. 448–453). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Ritter, A., Clark, S., Etzioni, O., et al. (2011). Named entity recognition in tweets: an experimental study. In *Proceedings of the conference on empirical methods in natural language processing* (pp. 1524–1534). Association for Computational Linguistics.

Röder, M., Usbeck, R., Hellmann, S., Gerber, D., & Both, A. (2014). N^3 - A collection of datasets for named entity recognition and disambiguation in the NLP interchange format. In *LREC* (pp. 3529–3533).

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1988). Learning representations by back-propagating errors. *Cognitive Modeling, 5*(3), 1.

Sánchez, D., Batet, M., Isern, D., & Valls, A. (2012). Ontology-based semantic similarity: A new feature-based approach. *Expert Systems with Applications, 39*(9), 7718–7728. doi:10.1016/j.eswa.2012.01.082.

Sawant, U., & Chakrabarti, S. (2013). Learning joint query interpretation and response ranking. In *Proceedings of the 22nd international conference on world wide web, WWW '13* (pp. 1099–1110). New York, NY, USA: ACM. doi:10.1145/2488388.2488484.

Shekarpour, S., Marx, E., Ngomo, A.-C. N., & Auer, S. (2015). Sina: Semantic interpretation of user queries for question answering on interlinked data. *Web Semantics: Science, Services and Agents on the World Wide Web, 30*, 39–51. Semantic Search. doi:10.1016/j.websem.2014.06.002.

Shen, D., Sun, J.-T., Yang, Q., & Chen, Z. (2006). Building bridges for web query classification. In *Proceedings of the 29th annual international ACM SIGIR conference on research and development in information retrieval, SIGIR '06* (pp. 131–138). New York, NY, USA: ACM. doi:10.1145/1148170.1148196.

Shen, W., Wang, J., & Han, J. (2015). Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering, 27*(2), 443–460. doi:10.1109/TKDE.2014.2327028.

Shen, W., Wang, J., Luo, P., & Wang, M. (2012a). Liege:: link entities in web lists with knowledge base. In *Proceedings of the 18th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1424–1432). ACM.

Shen, W., Wang, J., Luo, P., & Wang, M. (2012b). Linden: Linking named entities with knowledge base via semantic knowledge. In *Proceedings of the 21st international conference on world wide web, WWW '12* (pp. 449–458). New York, NY, USA: ACM. doi:10.1145/2187836.2187898.

Shen, W., Wang, J., Luo, P., & Wang, M. (2013). Linking named entities in tweets with knowledge base via user interest modeling. In *Proceedings of the 19th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 68–76). ACM.

Shi, B., Yang, L., & Weninger, T. (2017). Forward backward similarity search in knowledge networks. *Knowledge-Based Systems, 119*, 20–31.

Sil, A., & Yates, A. (2013). Re-ranking for joint named-entity recognition and linking. In *Proceedings of the 22nd ACM international conference on information & knowledge management, CIKM '13* (pp. 2369–2374). New York, NY, USA: ACM. doi:10.1145/2505515.2505601.

Sun, Y., Lin, L., Tang, D., Yang, N., Ji, Z., & Wang, X. (2015). Modeling mention, context and entity with neural networks for entity disambiguation. In *Proceedings of the twenty-fourth international joint conference on artificial intelligence (ijcai)* (pp. 1333–1339).

Turney, P. D., Pantel, P., et al. (2010). From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research, 37*(1), 141–188.

Tversky, A. (1977). Feaures of similarity. *Psychological Review, 84*, 327–352.

Usbeck, R., Ngonga Ngomo, A.-C., Auer, S., Gerber, D., & Both, A. (2014). Agdistis – Graph-based disambiguation of named entities using linked data. *13th international semantic web conference*. http://svn.aksw.org/papers/2014/ISWC_AGDISTIS/public.pdf.

Usbeck, R., Röder, M., Ngonga Ngomo, A.-C., Baron, C., Both, A., Brümmer, M., et al. (2015). Gerbil: General entity annotator benchmarking framework. In *Proceedings of the 24th international conference on world wide web* (pp. 1133–1143). ACM

Waitelonis, J., & Sack, H. (2016). Named entity linking in# tweets with kea. In *Proceedings of 6th workshop on neel challenge in conjunction with 25th WWW conference*.

Wu, Z., & Palmer, M. (1994). Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on association for computational linguistics, ACL '94* (pp. 133–138). Stroudsburg, PA, USA: Association for Computational Linguistics. doi:10.3115/981732.981751.

Zhang, L., Rettinger, A., & Philipp, P. (2016). Context-aware entity disambiguation in text using markov chains. In *Web intelligence (wi), 2016 IEEE/WIC/ACM international conference on* (pp. 49–56). IEEE.

Zhu, G., & Iglesias, C. A. (2017a). Computing semantic similarity of concepts in knowledge graphs. *IEEE Transactions on Knowledge and Data Engineering, 29*(1), 72–85.

Zhu, G., & Iglesias, C. A. (2017b). Sematch: Semantic similarity framework for knowledge graphs. *Knowledge-Based Systems*.

Zwicklbauer, S., Seifert, C., & Granitzer, M. (2016). Doser - A knowledge-base-agnostic framework for entity disambiguation using semantic embeddings. In *International semantic web conference* (pp. 182–198). Springer.