Location-Aware Named Entity Disambiguation

by

Maithrreye Srinivasan

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computing Science

University of Alberta

Abstract

Named Entity Disambiguation (NED) and linking has been traditionally evaluated on natural language content that is both well-written and contextually rich. However, many NED approaches display poor performance on text sources that are short and noisy. In this thesis, we study the problem of entity disambiguation for short text and propose a location-aware NED framework that resolves ambiguities in text with little other contextual cues. We show that the spatial dimension is crucial in disambiguating named entities and that the location inference is less utilized in many NED systems. Our proposed framework integrates (in an unsupervised manner) spatial signals that are readily available for many sources that emit short text (e.g., microblogs, search queries, and news streams). Our evaluation on news headlines and tweets reveals that a simple spatial embedding improves the accuracy of competitive baseline NED approaches from the literature by 8% for the news headlines and by 4% on tweets in probabilistic model. We further evaluated our spatial feature in a neural model and it showed that the NED performance is improved by 1.5% for the news headlines and by 6% for the tweets.

Preface

This thesis work is a collaborative effort with my supervisor, Dr. Davood Rafiei. While I performed the experiments, analyzed the results, and derived conclusions, Dr. Rafiei provided guidance and feedback on improving the approach, experimental design, presentation, and writing. A small portion of Chapters 4, 5 and 7 is published in the 30th ACM International Conference on Information and Knowledge Management [107] in 2021.

To my family for supporting me at each step

Acknowledgements

I would like to thank my supervisor, Dr. Davood Rafiei, for his continuous guidance throughout this research work. Without his support and insights, this work would have not been possible. I am so grateful for his constructive feedback, which has helped me to enhance my research skills. I enjoyed the freedom he gave me during my thesis work and his efficacious remarks helped me to think differently and find generic solutions to the problems.

I am so thankful to the members of the Database Research group. I learned immensely from the knowledge sharing sessions and it helped me to learn about different topics in research and stay relevant in the field.

I would also like to thank my friends for being a source of moral and technical support. I enjoyed various conversations about topics ranging from movies, technology to graduate life in general.

Last and most importantly, I would like to thank my parents for being a source of inspiration and motivating me to make living worthwhile. Without their belief and encouragement, I would not have come so far. I am grateful to my brother for his love, care and support.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Problem & Motivation

Names mentioned in documents and articles are often ambiguous, referring to more than one candidate entity, for example, in a knowledge base. For example, the name *"Michael Jordan"* represents more than ten persons in the English version of Wikipedia; some of which are shown below:

Michael (Jeffrey) Jordan, Former professional basketball player
Michael (I.) Jordan, an American researcher in Artificial Intelligence
Michael Jordan, Footballer
Michael (B.) Jordan, American Actor,
Michael H. Jordan, American executive for CBS, PepsiCo, Westinghouse.

The task of identifying these names, which are proper nouns mentioned in text, has been broken down into two crucial sub-tasks in Natural Language Understanding:

1. *Named Entity Recognition* (NER), which locates and classifies proper nouns to some of the pre-defined categories such as a person, organization, location, time expression, quantities, etc.

2. *Named Entity Disambiguation* (NED), which links the ambiguous mentions to a unique entity in the Knowledge Base (KB).

As another example, consider the following headline:

```
Smith reunites with Fresh Prince cast ahead of show's reboot.
```

In the headline above, who does "Smith" refer to? Does it refer to Sam Smith, an English Singer, or is it referring to Will Smith or Jaden Smith, both actors? In this particular case, knowing that "Fresh Prince" is a series acted by Will Smith, we can conclude that "Smith" refers to *Will Smith*. Both "Smith" and "Fresh Prince" are named entities referring to the names of a person and a series respectively, and the task of linking "Smith" to a unique entity "Will Smith" in a KB is called NED. The combined task of recognizing and disambiguating named entities is called Entity Linking (EL). These sub-tasks are essential components of many automatic language understanding tools, including semantic search [9], [109], knowledge base and knowledge graph population [55], [86], question answering [105], and chatbots [101]. The term NED and EL are used interchangeably throughout this thesis.

Resolving name ambiguities has always been a challenge with name variations and aliases, abbreviations (e.g., *JFK* referring to *John F. Kennedy*), spelling errors, polysemy (e.g., *John Smith* may refer to *American Actor* or *New Zealand Cricketer*), metonymy (e.g., *The White House* can refer to *American Administration*), etc.

Many NED systems resolve the ambiguities of mentions using various local and global features. The *local approaches* mainly focus on the lexical/syntactic features that include similarity between a mention and a candidate entity, surrounding words of the mention in the document, entity type (e.g., person, organization), prior probability, etc. They disambiguate each mention independently and fail to distinguish between two mentions with the exact surface text or shared context. For example, one cannot detect if the mentions of *Walker*, the last name, refers to *Casey Walker* or *Herschel Walker*, both professional football players.

To overcome some of the limitations of the local approaches, *global approaches* are introduced based on the assumption that mentions from the same

document are semantically coherent around the topic of that document [39], [41], [88]. The idea is to collectively perform disambiguation on all the mentions in the document, unlike the local approach where a single mention is disambiguated at a time. Some form of semantic similarity between the mention and the candidate entities may be used to measure the coherence score between mentions in the document. Incorporating coherence score or semantic relatedness has increased the accuracy in many NED systems. However, to calculate coherence for one entity mention, the NED system should be aware of mapping entities for other entity mentions in the same document. According to the work [43], [51], [61], the optimization problem of finding coherence is shown as NP-hard[1]; hence approximation algorithms and heuristics may be used to solve this problem, which is still a computationally intensive process in a NED task.

Global features fail when the input text is sparse. Sparse text lacks contextual information for these global features to work, and the prior probability does not always give a correct mapping. State of the art NED systems (e.g. [39], [51], [75]) mainly target long text, and many of these systems do not perform well in disambiguating entities in short text due to the lack of context. As an example, consider the following two short texts; *T1* is a headline from the NYT corpus [98] and *T2* is a text from a tweet collection [26].

(T1) Smith back in action after recovery
(T2) He's a Spurs lad, and we can't blame him for this season...

Passing these short texts to different NED systems results in different mappings of the named entity mentions, probably due to different prior probabilities of each system. For example, in text *T1*, "Smith" is mapped to *Tommy Smith*, a New Zealand Footballer born in 1990, by one system [51], *Agent Smith*, a fictional character from the movie The Matrix, by another system [33],

[1]A problem X is NP-hard if there is an NP-complete problem Y, such that Y is reducible to X in polynomial time. For example, the halting problem is an NP-hard problem where given a program and input; we determine whether the program will finish running or continue to run forever.

Adam Smith, a Scottish Economist, by a third system [79], and *Will Smith*, an American actor, by a fourth system [115]. Similarly, in text *T2*, "Spurs" is incorrectly mapped to *San Antonio Spurs* by multiple systems [33], [51], [79], [115]. However, knowing that headline T1 is originated or published in Sydney, Australia, one may say with some confidence that the mention of "Smith" refers to *Steve Smith*, an Australian cricketer. Similarly, if we take into consideration the fact that tweet T2 is posted by someone from London, England, we may link "Spurs" to *Tottenham Hot Spur F.C.* and not *San Antonio Spur*.

(a) (b)

Figure 1.1: Signature of Candidates (a) "Spurs" and (b) "Smith".
(a) Purple Marker denotes location Signals of "San Antonio Spurs" and Green marker denotes "Tottenham Hotspur F.C"
(b) Purple Marker denotes location Signals of "Steve Smith" and Green denotes "Will Smith".

Our goal in this thesis is to explore additional cues that are readily available for short text in the form of metadata. In particular, tweets and news headlines possess rich metadata information, notably temporal and spatial data, often recorded and emitted by capturing devices (e.g., mobile phones and GPS-enabled cameras). On the other hand, the spatial signal is a crucial property not just of physical entities such as countries, mountains, and rivers but also of persons, organization headquarters, artifacts, and events such as sports leagues, battles, etc. For example, *Steve Smith* was born in *Sydney, Australia* and *Tottenham Hot Spur F.C.* is a football team in *Tottenham, London*. These spatial signals for candidate entities may be collected from relations such as *happendIn, isLocatedIn, isplacedIn, bornIn, diedIn* in a knowledge base or a

knowledge graph. Not all entities are popular around the globe. From Figure 1.1, we can see that *San Antonio Spur* is popular in the US and *Tottenham Hot Spur F.C.* is popular in the UK, and that knowing the location of an ambiguous mention can help the NED process. In this thesis, we explore the location cues available for short text and study the problem of modelling all the location information associated with a mention or an entity for enhancing the disambiguation decision. Our approach to this problem is to compute location signatures for entities from different sources where entities are mentioned (e.g. tweets) or discussed (e.g. Wikipedia pages) and use the spatial dimension as an expressive feature when comparing candidate entities against the context of an input mention for linking the mentions to its correct canonical form.

1.2 Thesis Statement

The research hypothesis is that spatial cues provide a rich context for named entity disambiguation and that such cues are quite useful in disambiguating named entities in short text which lacks detailed contextual information.

1.3 Contribution

Our contribution can be summarized as follows:

- We propose a framework for integrating spatial signals in disambiguating named entities in short text. To the best of our knowledge, our work is the first studying the problem in the context of short text.

- As most of the entities have a relationship with multiple locations beyond their primary home location, we develop an algorithm to construct location signatures for entities and context mentions. The spatial signatures are embeddings that reflect the importance of different locations associated with the entities. They are automatically created by extracting and normalizing spatial expressions in entity descriptions such as Wikipedia articles. Similarly, spatial signals are captured in the context of textual mentions and represented by embeddings.

- We evaluate our model for short text on two data sets - headlines from the New York Times archives and Tweets with rich geographical information. Our work significantly improves, in terms of accuracy and the F1 measure, several baselines from the literature, including some considered the state of the art.

- We extend our work by evaluating spatial signatures in the context of a neural approach using the distributed representation of mentions and entities and show that a slight improvement in the accuracy and F1 measure is still possible despite the richness of the neural approach.

1.4 Outline

The rest of the paper is organized as follows: **Chapter 2** discusses components involved in knowledge base population and similar problems like entity disambiguation in resolving the ambiguity of mentions. **Chapter 3** discusses different NED baselines and state-of-art systems in the literature. **Chapter 4** introduces the geospatial dimension of named entities. **Chapter 5** and **Chapter 6** discuss our probabilistic framework and neural framework of NED for short text. **Chapter 7** showcases the detail of the dataset and evaluation of probabilistic and neural framework. **Chapter 8** summarizes conclusions and possible direction for future work.

Chapter 2

Background

Named Entity Disambiguation is usually a part of the more extensive pipeline that may include other components such as information extraction and knowledge base population. This pipeline or part of this pipeline can be used in different applications like question answering. In this chapter, we provide some background on some of these components and their relationship to NED and see how one or more components are used in question-answering system. We will also provide background about two closely related problems to NED in resolving the ambiguity of the mentions. As illustrated in Figure 2.1,

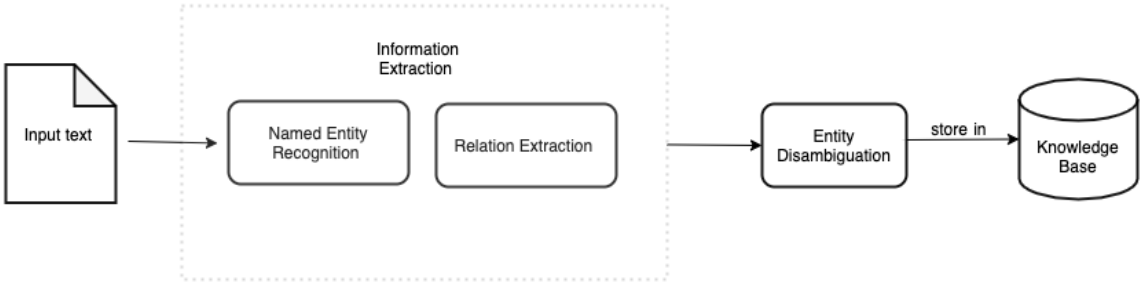

Figure 2.1: General Framework for Knowledge Base Population System

a knowledge base population system first identifies named entities from unstructured sources like Web documents and extracts their relations through a relation extraction technique. Then, it performs NED to populate the extracted knowledge into a KB.

2.1 Information Extraction

Information Extraction (IE) is a module that often precedes or encompasses NED and refers to the task of extracting structured information about entities from unstructured textual sources and storing it in the database. The extracted information may include facts and relations between entities, and such information may be used in intelligent content classification, integrated search, mining for patterns and trends, uncovering hidden relationships, and knowledge discovery. The two important sub-tasks of IE are *Named Entity Recognition* and *Relation Extraction*.

2.1.1 Named Entity Recognition

Named Entity Recognition (NER) identifies the named entities from a text document and classifies them to some predefined categories such as person, organization, location etc. In the following example, the underlined words are named entities,

> Mark Zuckerberg is one of the founders of Facebook, a company from the United States.

The words Mark Zuckerberg may be labelled as *person*, Facebook as *organization* and the United States as *location*.

In the early 90s and 00s, MUC-6 [40], and CoNLL shared task [99], were introduced to help with the development and evaluation of identifying named entities and assigning them to coarse entity types, distinguishing between the types of person, organization, location and miscellaneous. NER is a sequence labelling problem that may use a supervised machine learning approach for label prediction. Given an annotated sentence with IOB tags (Inside, Outside, and Beginning of a named entity), a classifier is trained to tag words to identify them as named entities in the sentence. Stanford NLP [35], a well-known NER tool, uses Conditional Random Field (CRF) with Gibbs sampling to classify the labelled sentence into the person, location, organization and miscellaneous. The authors use the POS tag and the surrounding POS tag sequence, current

word n-gram, words in left and the right context, current, previous and next word as features and train a model on datasets like CoNLL, MUC-4, MUC-7, and ACE Corpora.

Fleischman et al. [36] address the limitations of coarse-grained entity types and introduce more fine-grained sub-classes for a person like an athlete, politician/government, clergy, businessperson, entertainer/artist, lawyer, doctor/scientist, and police. Rahman et al. [93] increase the granularity of types to 92 fine-grained semantic classes such as date, time, money, quantity, nationality, religion or political group, etc. However, their model is limited in assigning one type per mention. Nakashole et al. [80] and Lin et al. [65] are some of the work that assigned multiple types to mentions using fine-grained type hierarchies of Freebase and YAGO.

Starting from Collobert et al. [18], Deep learning based NER with minimal feature engineering have been flourishing [54], [62], [89]. One such model, ACE (Automated Concatenation of Embeddings) proposed by Wang et al. automatically searches for better embedding concatenation in structure prediction tasks. The model uses a simple search space and reinforcement learning with a novel reward function to efficiently guide the controller to search for better embedding concatenations [113]. This model is the current state-of-art in the CoNLL 2003 NER task.

2.1.2 Relation Extraction

Relation Extraction (RE) is the task of detecting and extracting semantic relations between the named entities from text. Each extracted relationship occurs between two or more entities and falls into one of several predefines relations such as *is_in*, *located_in*, *play_for*, *employed_at*, *founded_by*, etc. Several EL systems harness the coupling of lexical types for entities with the type signature for relations to distinguish between ambiguous entities [81], [94]. For example, consider the following input sentences:

(1) Amy received the Grammy for the best new artist.
(2) Amy received her degree in neurobiology from Harvard.

Amy, Grammy and Harvard are named entities and received_award, received_degree_from are relations. To properly distinguish the entities *Amy_Winehouse* and *Amy_Farrah_Fowlerin* in the above examples, understanding the different type signatures of the relations *received_prize* (Singer $\times$ Music award) and *received_degree_from* (Person $\times$ University) is very crucial. In our work, we utilize the location cues lodged in one or more relations like *locatedIn*, *isCaptialOf*, *wasBornIn* etc., and distinguish ambiguous mentions based on their spatial similarity.

RE techniques were broadly classified into one of the following categories: Supervised, Distant supervision, Semi-supervised, Unsupervised and Open Information Extraction (OpenIE). In a supervised approach, a standard binary classifier may be trained to predict whether there is a specific relation between entities and label them [78], [83], [121], [122]. This approach gives high precision but involves manual annotation, which is expensive in cost, effort and time. This has led to some of the semi-supervised techniques like Bootstrapping [7], [11], active learning [117], label propagation methods [16]. RE has also been widely studied using distant supervision as it combines the advantage of both supervised and unsupervised paradigms [66], [76]. Another widely studied area is OpenIE, which aims to extract relational facts on an open-domain corpus, where the relation types may not be pre-defined. However, OpenIE identifies relations based on textual surface information or heuristics[23], [70]. Unsupervised relation extraction methods have not been explored as much as fully or distantly supervised learning techniques. An unsupervised approach can discover new relation types, since it is not restricted to specific relation types in the same way as fully and distantly supervised methods [104].

2.2 Knowledge Base Population

Knowledge Base Population is the task of populating or completing the incomplete elements of a knowledge base from unstructured sources. EL is considered an important subtask for the knowledge base population. Given a relation or fact which is needed to be populated into KB, if the entity associated with the relation has its entity record in KB, then EL takes place. Therefore, populating a knowledge base can potentially benefits from entity linking. Besides EL, slot filling which fills in values and relations of given entities with facts extracted using IE, event tracking which extracts information about events, beliefs and sentiments tracking about entities are other sub-tasks that are combined with NED to populate a KB with the extracted information from text.

As the world is constantly changing, so are the new facts that are surfacing. Keeping up with these changes requires a continuous effort, thus becoming a demanding task at scale. Recent advancements in IE have led to an automatic knowledge population (e.g. NELL [15]). Existing KBs such as Cyc [64], DBpedia [3], Freebase [10], WordNet [29] are constructed manually or through crowd sourcing communities. Wikipedia, YAGO [50], DBpedia and Freebase are some of the KBs that have been widely exploited in entity linking including our work which uses Wikipedia and YAGO.

2.3 Question-Answering Applications

A typical task that requires NED to help resolve the ambiguity of named entities is question answering (QA). It is the task of generating answers to user questions that may leverage a KB [105]. To get a correct answer, we must place the question semantics through the mentions and relations available in KB. So the first step in the QA is to identify the mentions in the question and link them to entities in KB. As shown in Figure 2.2, we see two entity mentions that are detected and are linked to the KG referents (Taylor swift, Album), and the extracted relations are also detected of types instanceOf and performer. EL is an essential step for QA as it leads to a correct answer via

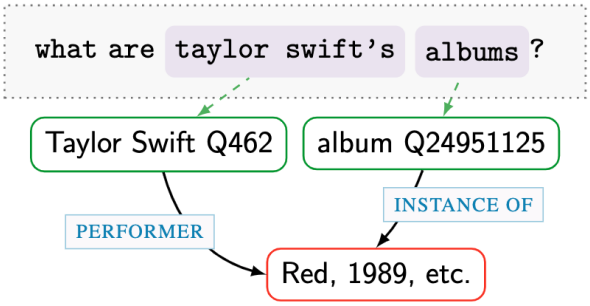

Figure 2.2: Example question from a QA dataset that shows mention detection and their relationship with correct answer to the question. Figure is taken from Sorokin et al. [105]

some connected paths (relations) between the named entities mentioned in the question.

2.4 Other forms of reference resolution

The following section gives a brief overview of task that resembles NED, namely coreference resolution and word sense disambiguation.

2.4.1 Coreference Resolution

When performed without KB, EL can be reduced to a coreference resolution problem. Coreference resolution is the task of finding all expressions that refer to the same entity in a text. The referent can be a noun phrase, a named entity or a pronoun. Coreference resolution is an important step in obtaining unambiguous sentences which computers can understand more easily. The difference between NED and Coreference resolution can be seen with the following example

> **Michael Jackson** is regarded as one of the most significant cultural figures of the 20th century. The **King of Pop** won eight Grammy Awards in one night in 1984.

NED will pick Michael Jackson, King of Pop and Grammy Awards as men-

tions and tries to find a candidate list for each mention. In this case, Michael Jackson and King of pop are treated as two different mentions but might be linked to Micheal Jackson at the end. However, a coreference resolution model will link King of pop to Micheal Jackson. An overview of the recent work in coreference resolution is given in Zhang et al. [118].

2.4.2 Word Sense Disambiguation

EL is also similar to the problem of word sense disambiguation (WSD). WSD is the task to identify the sense of a word (rather than named entity) in the context of a sense inventory (e.g., WordNet) instead of a knowledge base. WSD regards the sense inventory as complete, whereas KB is often treated as incomplete. Though NED and WSD look superficially similar, the key differences are that named entities mentioned in the text can have a lot more candidates which can make them more ambiguous. For example, a common first name "Bob" has more than 1000 candidate entities in Wikipedia. Take the following sentences as an example.

(1) I can hear *bass* sound.
(2) They like grilled *bass*.

The occurrences of the word "bass" in the above two sentences denote different meanings: low-frequency tones in the first sentence and a type of fish in the second sentence. An overview of the recent works in WSD is given in Bevilacqua et al. [8].

Chapter 3

Literature Review

Named entity disambiguation has a long history with some early work on record linkage where the task is to find out if two records in a database represent the same entity [30]. The literature on NED is vast, with an extensive list of approaches. Shen et al. [102], and Sevgili et al. [100] provide a thorough overview and analysis of the main approaches of the EL system. This chapter will see a detailed literature review of different NED baselines and start-of-art (SOA) systems.

There are mainly three categories of approaches proposed in the literature: (1) disambiguate mentions individually using ranking (Section 3.1); (2) disambiguate mentions collectively by solving an optimization problem (Section 3.2) and (3) disambiguate mentions using a deep neural network (Section 3.3). We also present some NED works for short and informal text that use one or more of the previously mentioned approaches (Section 3.4).

3.1 Local NED

Early work on NED focuses on disambiguating mentions in input text document in isolation. These approaches often use a compatibility function $\Phi : MXE \rightarrow [0, 1]$ to measure the local compatibility between a mention m_i and each of its candidate entity $e_{ij} \in cand(m_i)$ with the goal of finding,

$$e^* = \underset{e_{ij} \in cand(m_i)}{\text{argmax}} \ (\Phi(m_i, e_{ij}))$$

Various features pertaining to the entity surface text, entity popularity and context are used in both unsupervised and supervised manners. In the following subsections, we review a few proposed methods from the literature using the compatibility function.

3.1.1 Unsupervised Approaches

A typical way to compute compatibility function is to model each mention and each entity as feature vectors and employ vector-based similarity measures. Bagga and Baldwin [4] were the first to measure the compatibility function using the vector space model by clustering all the references corresponding to the same entity. Bunescu and Pasca [13] followed a similar approach where they used the similarity of the context surrounding the mention with the entity Wikipedia article. The authors also used word-category association by correlating the words in the context of mention and categories of each entity. This word-category association helps to counter context-spareness issue that arises due to word variations (e.g. agreement vs agreed) and lack of semantics. Bunescu and Pasca [13] laid the foundation for creating a candidate dictionary by using redirects and disambiguation pages and for using Wikipedia links for the ground truth entity. Cucerzan et al. [19] also used named entities extracted from a document along with the surrounding window as the context in their work.

3.1.2 Supervised Approach

Many supervised NED approaches build classification and regression models replacing the compatibility function $\Phi(m_i, e_{ij})$ in an unsupervised approach. These approaches may use a regression model to predict a confidence score and use it to rank the candidate entities instead of using a binary classification to determine whether an entity e is a correct entity of a mention m or not. Supervised NED approach has some advantages. First, we can leverage the abundant data from Wikipedia for training. Second, it is easy to tune the parameters in a supervised approach which remains a challenge in an unsupervised approach.

Mihalcea and Csomai [73] use a simple Naive Bayes classifier to combine features such as mention surface text and their POS tag, a window around Wikipedia links and a list of frequent keywords in the context. The authors identify important concepts in the input text and automatically link these concepts to the corresponding Wikipedia pages, thus introducing word sense in the feature set (e.g. Plane can be aircraft or theoretical surface of infinite area). Milne and Witten [75] do a Wikification[1] similar to Mihalcea and Csomai. The authors train several classifiers such as naive bayes, support vector machines (SVM) and few decision tree algorithms, for disambiguating named entities. Of these classifiers, a variation of decision tree outperforms the rest of the classifiers. The feature set contains three features: commonness or prior probability, semantic relatedness between entities and context quality which measures the weight of the context (computed from both commonness and relatedness). Dredze et al. [21] proposes SVM based learning algorithm that uses 55 classes of features that includes features from mention such as name matching, string equality, acronyms, entity popularity, input context, entity types and KB statistics, KB categories etc. They have achieved 94% accuracy on the newswire dataset and 80% on the TAC-KBP dataset. Zhou et al. [124] employ two learning algorithms Gradient Boosted Decision Trees and Gradient Boosted Ranking, and use 20 types of semantic relatedness scores between entities in addition to features extracted from the documents and KBs. They achieve 84% accuracy on MSNBC and 81% on Yahoo! news dataset.

3.2 Global NED

A global NED exploits the coherence of the mentions in a given document by collectively disambiguating the mentions. The topic coherence assumption aims to find an assignment Γ with the maximum global coherence among the entities in Γ. These approaches are formalized either as optimization

[1]The process of adding links to Wikipedia to specific words and phrases in an arbitrary text.

problems or dense sub-graph problems. Suppose $\Psi : \Gamma \Rightarrow \mathbb{R}$ is a measure of global coherence of an assignment Γ, then NED can be cast as an optimization problem aiming to find an assignment for Γ such that:

$$\Gamma^* = \underset{\Gamma}{\operatorname{argmax}}(\sum_{i=1}^{N}\sum_{j=1}^{k}\Phi(m_i, e_{ij}) + \Psi(\Gamma))$$

Here N is total number of mentions in the document, $\Phi(m_i, e_{ij})$ measures the compatibility score of the mention m_i with its jth candidate entity e_{ij} where $j = 1, ..., k$ with k being the total number of candidate entity. $\Psi(\Gamma)$ measures the global coherence. The above problem will become a local ranking problem without the $\Psi(\Gamma)$ component.

Cucerzan et al. [19] propose a global NED approach in which semantic relatedness is measured using the overlapping of categories of entities from Wikipedia. Their evaluation of the MSNBC news dataset shows significant improvement over local NED approaches. Kulkarni et al. [61] propose NED as collective inference solution with local spot-to-entity compatibility and page-level topical coherence features. The authors train supervised models on Wikipedia to measure the local compatibility and employ a semantic relatedness measure based on inlinks of entities used by Milne and Witten [75], to measure the global coherence. The authors use heuristics based on local hill-climbing and linear program relaxation for a collective inference and their work outperforms Cucerzan [19], Milne and Witten [75] and Mihalcea and Csomai [73] on both MSNBC and IITB. Ratinov et al. [95] treat NED as an optimization problem with a two-stage approach where an SVM ranker chooses the best candidate entity, and an SVM linker predicts if the selected candidate is the ground truth entity or not. The feature set includes the disambiguation confidence and link-likelihood from Wikipedia, the context of entities taken from top 200 tokens from Wikipedia weighted by tf-idf, the context of the mentions from their document and the semantic relatedness defined using both the Pointwise Mutual Information (PMI) and a variation of Normalized Compression Distance on the inlinks and outlinks of entities. Their approach outperforms Milne and Witten [75].

Cheng and Roth [17] propose a model that explores the relational constraints in both in candidate generation and disambiguation stage. They formalize NED as an integer linear programming problem to find an assignment that maximizes coherence and relational inference. Relational constraints are added through syntactico-semantic relations by leveraging the relational triples from Wikipedia and DBpedia (e.g. the mentions "Iran" and "Ministry of Defense" in an input text will help to resolve to the correct entity "Iranian Ministry of Defense" using the relation between these mentions) and co-reference relations. Their method improves accuracy and outperforms various previous methods [75], [95] by improving coherence using relational constraints.

Hoffart et al.'s AIDA [51] cast NED as a dense subgraph problem where mentions and candidate entities are nodes and mention-entity and entity-entity relations are edges. The mention entity relation is defined in terms of the local context similarity, and the entity-entity relation is given by semantic relatedness. The goal is to find a subgraph with the lowest sum of weighted edges containing all the mentions and entities with one-entity-per-mention constraint. AIDA outperforms Kulkarni et al. [61] and remains the state of the art during its time. Our work uses the prior probability and keyphrase weights used by AIDA [51]. Similar to AIDA, Han et al. [47] propose a graph-based collective entity linking method exploring the global interdependence between the candidate entities in the dense subgraph. The proposed method performs a random walk on the subgraph by enforcing interdependence between NED decisions using the iterative evidence propagation method. The result of NED decisions is propagated to the other nodes in the graph, and an entity is chosen as a true entity if it maximizes the similarity of the local and global compatibility. This method slightly improves global models with interdependence in a pair-wise fashion [61], [75]. Guo et al. [42], [43] use a random iterative walk with a restart to find the semantic signature of entities. The model constructs an entity graph like other global approaches and represents each candidate entity by the stationary probability distribution resulting from a random walk on that graph. The semantic representation uses more relevant entities from the knowledge graph, thus reducing feature sparsity and resulting in substan-

tial accuracy gains. The semantic relatedness is measured using a variation of Kullback-Leibler (KL) divergence called zero-KL divergence to handle the case of $Q = 0$ given two probability distribution P and Q.

3.3 Neural NED

Neural networks show potential in various NLP tasks, including the NED task. Deep neural networks can learn sophisticated representations within their deep layered architectures. Neural network models can reduce the burden of manual feature engineering and may enable significant improvements on EL and other tasks.

He et al. [48] were the first few to apply deep neural network (DNN) to the NED task. They propose a stacked de-noising auto-encoder to learn a continuous representation of both the context and the entity documents in an unsupervised pre-training stage. A supervised fine-tuning stage follows the pre-training stage to optimize the representation and to learn similarity measures for local compatibility. The experiment uses TAC KBP and AIDA datasets and displays the state-of-the-art performance beating a few collective global approaches using only local compatibility. Sun et al. [108] build a variable-sized context model with a convolution neural network (CNN) and embeds the positions of context words to factor in the distance between context words and mentions. Similarly, Francis-Landau et al. [37] use a CNN to capture the semantics for the mentions and entities in a continuous vector space. The convolution layer operated on three different granularities of semantics; mention, surrounding text and the document containing the entity mentions to capture various kinds of topic information. These are then combined with a sparse linear layer to achieve a state-of-the-art performance on some entity linking datasets such as CoNLL, ACE, etc. Nguyen et al. [82] extend the work of Francis-Landau et al. [37] by using recurrent neural network (RNN) to enforce topic coherence.

Yamada et al. [115] propose a new embedding method extending the skip-gram model [74] for NED. The Skip-gram model is extended in two aspects:

the KB graph and the anchor context model. The KB graph model learns the relatedness of entities using the link structure of the KB, and the anchor context model aims to align similar words and entities in the vector space by leveraging KB anchors and their context words. Their NED approach using both local and global compatibility achieves state-of-art accuracy on CoNLL-AIDA and TAC datasets. Gupta et al. [44] propose a model that learns a dense representation for each entity from various sources of information such as its entity description, contexts around mention and its fine-grained types. Phan et al. [92] improve this approach using two long short-term memory (LSTM) to inject positional information of mention and the words in the embedding. They also add an attention mechanism to handle noises in context. The experimental result shows that positional information and word order can improve the accuracy by 5% to 10% on their evaluation dataset.

Raiman et al. [94] proposes DeepType, a model that integrates symbolic information into the reasoning process of neural networks. They construct a type system and use it to train the neural network to respect the symbolic structure. DeepType outperforms all existing solutions by a wide margin on WikiDisamb30, CoNLL and TAC KBP 2010 datasets. Similarly, Onoe et al. [84] proposes a method using fine-grained entity properties to disambiguate closely related entities.

Ganea and Hoffman [38] propose a model that combines (1) entity embeddings, (2) contextual attention mechanism, (3) an adaptive local score combination and (4) unrolled differential message-passing algorithm for global inference. This model standouts for training entity embeddings from scratch and paves the way for the subsequent neural models to use their entity embeddings. The authors train their entity embedding by bootstrapping from their canonical entity pages, and local context of the hyperlink annotations in contrast to the models producing entity embeddings using entity-entity co-occurrences (suffering from sparsity issue) [115], [125].

Kolistas et al. [59] propose the first end-to-end EL model that jointly discovers and links entities in a text document. The proposed model learns a context-aware mention embedding from word embeddings using bi-LSTMs

and uses the entity embeddings from Ganea et al. [38] to measure the context similarity between a mention and an entity. The context model in our neural framework is inspired by Kolistas et al. [59]. Similarly, Fang et al. [27] use LSTMs to encode contextual word sequences and find a local similarity between the contextual word embedding and entity embedding using multilayer perceptron. The author also uses LSTMs to enforce a global coherence and maps the disambiguation problem into a reinforcement learning problem. The model sequentially learns a policy to select target entities and makes decisions based on the current and previous states.

Logeshwaran et al. [69] propose a zero-shot baseline using a pre-trained reading comprehension model and show that attention between a mention in context and entity descriptions is critical to the generalization ability of entity linking systems with minimal assumptions. They use domain-adaptive pre-training for domain-shift problems present in the EL for unseen entities. Similarly, Wu et al. [114] propose a two-stage approach for zero-shot EL based on a fine-tuned BERT architecture. The retrieval in the first stage independently embeds the mention context and the entity descriptions. In the second stage, the proposed model uses the cross-encoder to look through the candidate entities and concatenates the mention and entity text following Logeshwaran et al. [69]. Their work improves the accuracy of the prior global models on the TACKBP-2010 dataset [94], [103], [115] and the normalized accuracy of [69].

Ravi et al. [96] propose CHOLAN that focuses on improving the performance of the EL system using a transformer-based model. Their experiment shows that transformers require additional task-specific context to improve the EL performace. CHOLAN out-perform the state-of-the-art Kolistas et al. [59] as the accuracy of the previous transformer-based EL model's [12], [34], [90] were lower than Kolistas et al. [59].

3.4 NED for short text

In the past few years, Twitter and other microblogging sites have received increasing attention as they are a rich source of information for a wide variety of

topics. A large number of previously proposed NED system address Wikipedia-style text and news corpus. However, the performance drops drastically when applied to short text. Hence, it is important to know the EL methods that are specifically designed for short text such as news feeds, tweets, chat transcripts, search queries, etc. The following provides an overview of the EL system in the literature for short text in supervised, unsupervised and neural settings.

TagMe by Ferragina and Scaiella [33] is the first system to perform NED on short text. They provide on-the-fly annotation by performing a voting scheme that computes a scoring function for local compatibility and coherence without any expensive collective inference. This model outperforms the best-known system at that time [61], [75] for short text and provides competitive results on long text within a news corpus. Babelfy proposed by Moro et al. [79] combines word sense disambiguation and entity linking by running a random walk model that assigns weights to the edges of a semantic network and chooses the densest subgraph for the semantic interpretation. The choice of a dense subgraph is expected to give the most coherent meaning to multiple interpretations. Falcon by Sakor et al. [97] is a framework that leverages fundamental principles of English morphology like compounding and headword identification and jointly performs entity and relation linking using DBpedia as KG. Falcon outperforms Babelfy [79], KEA [112], AIDA[51], TagMe[33] for EL on the question-answering dataset(QALD-7 and LC-QuAD).

Meij et al. [71] propose a model that uses n-gram and other tweet features for linking tweets to Wikipedia articles based on the concept ranking. First, they rank their candidate entities based on high recall and then use a machine learning algorithm (SVM, RF, GBRT) to rerank them to achieve the best accuracy. The concept ranking model shows significant improvement compared to [72], [75]. Guo et al. [41] employ a structural SVM algorithm for jointly detecting mention and disambiguating entities in tweets as a single end-to-end task. They use capitalization rate, entity type, prior, tf-idf score and other semantic features in their feature set and use voting strategy by TagMe [33]. Their evaluation outperforms TagMe [33] and Cucerzan et al [19]. KEA by Waitelonis and Sack [112] uses surface text similarity, DBpedia types, and a co-

occurrence analysis of mentions within Wikipedia articles for disambiguation. It employs different techniques to decide which candidate is chosen as the winner for a mention. The most basic approach considers the weighted sum of the scores as a confidence score, whereas the weights are optimized via grid search on a given development or training dataset. Yang et al. [116] propose S-MART, a tree-based structured learning framework based on multiple additive regression trees for EL on a Tweet dataset [26]. The feature set includes base, capitalization rate, popularity, context capitalization and entity type categories.

Feng et al. [31] propose a Knowledge Enhanced NED model that creates entity context embedding from word2vec and conceptual embedding from a KB. The self-attention mechanism maps the candidate entities and the context to the same semantic space to find the best candidate entity that semantically matches the context. Eshel et al. [24] employs a neural model that uses a gated recurrent unit and an attention mechanism to capture limited information around the short and noisy text. They use Yamada et al. [115] word and entity embedding but initialized embedding using skip-gram with a negative-sampling algorithm. The authors use corrupt sampling to generate negative examples. Their method improves the accuracy by 5% compared to Yamada et al. [115] on the WikiLinks dataset which is 3.2M short fragments from different web pages. Huang et al. [52] propose a model that uses pre-trained BERT embedding and represents entity candidates using information from a KG such as entity name labels, entity description and the relations between entities. Their result shows improvement compared to S-MART[116], VCG[105] and DBPedia Spotlight.

Few works use Spatio-temporal features for NED on short text like news headlines and tweets. In particular, Fang and Chang [26] use Spatio-temporal signals in a weakly supervised fashion for linking entities mentioned in tweets. They use a binning method to divide both time and space into discrete equal-sized bins. Agarwal et al. [1] use the publication year of documents to build a temporal vector for each entity and show that the temporal feature helps in disambiguating mentions in a short text. Our research work is closer to

this [1], [26] as we use spatial signals from the input text and KB to enhance the NED for short text.

Chapter 4

Geo-Spatial Dimensions of Named Entities

4.1 Spatial signals as feature

Real-Time Location Systems and Location-Based Services have shown tremendous growth in recent years as spatial data analysis has become highly essential for companies to understand their business trends across regions. For example, Amazon and Netflix offer location-based recommendations of their products and movies. In times of disaster, microblogging services like Twitter and Facebook provide real-time awareness by brokering information about the conditions on the ground and the status of relief efforts, as well as providing a platform to express concerns about the relief effort. In all these applications, location data is used in real-time with traditional data to gain better insights.

All physical entities have a geospatial footprint in space [50]. Named entities in the KB such as countries, cities, townships, mountains, and rivers usually have a permanent location which may be represented as text, geo coordinates, polygons, and bounding boxes. These spatial data are available in KB (e.g. Wikipedia, Yago) under different relation types. Events such as battles and sports leagues may be recorded under *happendIn* relationship in a KB; For example, the festival of San Fermin (Running of Bulls) happens in Pamplona, Navarre in Northern Spain. Groups or Organizations with the venues or headquarters are recorded under *isLocatedIn* relation. Artifacts are physically located at a particular place; for example, the Statue of Liberty is

located in the harbour of New York City.

The spatial dimension of entities may also be expressed in the form of relations. For example, Barack Obama was *born in* Honolulu, Hawaii; Djokovic won the Grand Slam tennis singles in the 2021 Wimbledon Championship *held at* London.

On the other hand, each location has a rich spatial relationship, such as containment and neighbourhood with other locations. For example, suppose a mention or an event is tied to *Edmonton*. In that case, we can say that it is tied to *Alberta* and *Canada* because of the containment relationship between their location. Similarly, knowing that the event happens in Edmonton has implications for the neighbouring locations such as *Leduc*, *Nisku* etc.

To resolve locations, we need a geographical database (aka gazetteers) that can provide rich location information such as geo-coordinates, hierarchy, alternate names, etc. Wikipedia has around 200,000 geo entities, but not all the geo-entities in Wikipedia are associated with geo-coordinates. GeoNames and Open Street Map are well-known gazetteers which are extensively used in many location-based services and applications. We use GeoNames[1] containing over 25 million geographical names, ISO country codes, alternate names, latitudes and longitudes, area and population statistics. It also provide the hierarchy of locations given in a tree structure, starting with the world as the root and broken down to different levels of dispersion such as continent, country, state and city.

4.2 Ambiguity of locations

There are some challenges in using spatial information as a feature. In particular, location references can be ambiguous and such ambiguity can be categorized into two classes.

1. Geo/Non-Geo ambiguity
2. Geo/Geo ambiguity

[1] www.geonames.org

The Geo/Non-Geo ambiguity is when a named entity can be both a location and a non-location entity. For instance, a mention of *Sharon* can be the name of a person or a place, e.g. a city in Massachusetts. Geo/Geo ambiguity is when mentions identified in text may refer to more than one location. For example, the term *Paris* by itself refers to 34 different places around the world (e.g., Paris, France; Paris, Ontario; Paris, Arkansas).

4.3 Methods to handle location ambiguities

For each mention of a possible location (e.g. tagged by NER tool), we may search the mention's text in a geographical database to get the list of potential matches. If no results are returned, we may treat the mention as a non-location word. To handle Geo/Non-Geo ambiguity, one approach is to use spatial indicators such as the preposition "in" in a syntactic construct to distinguish the noun phrases that are geographic locations from non-geographic references [85].

To handle geo/geo ambiguity, we may leverage the containment relationship between different mentions in the same text. The intuition is that a canonical location can be mentioned or referred to directly by its name or by the name of a location in a higher granularity such as province, state and country. When no such sign at a higher granularity is present, we may use other information about a location such as population [2], [56] and disambiguate a name to a location with a higher prominence. For example, consider a page that mentions "Paris"; it can be Paris, France or Paris, Ontario. Suppose the page also mentions "Canada" or "Ontario", in that case, "Paris" here likely refers to the city in Ontario, Canada. When there is no additional information at a higher granularity level, Paris can be resolved to Paris, France, as it is more populated and is known by more people than other candidates.

Linking location references to a geographic database (e.g., Geonames) is studied on its own in the literature (e.g., [57]) and is outside the scope of this thesis. So any suitable method from the literature may be used. One may also want to calculate a term distance [56], [85] or a confidence score [2] to

decide which location-mention and near mention should be considered for a resolution.

4.4 Spatial signature of entities

Given an entity page in a KB (e.g. Wikipedia), the spatial attachment of the entity to a location may be assessed by the mentions of the location in the entity page. To construct a spatial signature for an entity, we may use the locations tagged by NER. The number of times a location occurs in an entity page can give a degree of association between the entity and the location. Hence, we tag all locations and get the frequency count of each tagged location in the entity page.

Let the initial location signature of an entity e, denoted by l_e^c, include all locations that are explicitly mentioned in the entity page of e and their frequencies c. One may do a spatial smoothing to account for locations that are relevant but not listed on the Wikipedia page of an entity. This is useful in news pieces and micro-posts where neighbourhood events are reported. Lets take the following example,

> Teen violent assault at a Catholic high school south of Edmonton

This is reported in *Edmonton*, but the actual event takes place in *Leduc*, a neighbourhood of Edmonton. As Edmonton is a geopolitical place, the neighbourhood is indicated indirectly through Edmonton. To handle such a case, we get the neighbours of the mentioned location from a gazetteer. In particular, we query for the sibling relations of the mention in Geonames since the mention share the same higher level of dispersion (e.g. province and country) and the same siblings administrative level and feature class. In the case of "Edmonton", its neighbours "Leduc" and "Nisku" are in the same province as Edmonton. Here Leduc is a relevant location, but it's not listed in the context.

Let l_e^s include all locations in l_e^c with their counts and all other locations that

are relevant but not in l_e^c with counts set to zero. The relevant locations of l may include its siblings and ancestors in a location hierarchy. One may provide equal weights to all the neighbours of a location. However, not all neighbours may help with the disambiguation and nearby neighbours are usually more relevant. For example, "Red Deer", roughly 130 km away from Edmonton, is a sibling of Edmonton sharing the same parent and feature class. However, the intuition is that both the authors and the readers of a news piece may refer to "Leduc" as "South of Edmonton" due to its closer vicinity to "Edmonton" but it is less likely that "Red Deer" is referred the same.

With the intuition that closer neighbours are expected to contribute more to the confidence score of an entity disambiguation, we use exponential smoothing as spatial smoothing for unseen neighbours. Exponential smoothing is a common technique for smoothing time series data using an exponential window function which assigns exponentially decreasing weights over time. This is one of the many window functions commonly applied to smooth data in signal processing and to remove high-frequency noise. The simplest form of exponential smoothing s_t is given by

$$s_t = \alpha x_t + (1 - \alpha) s_{t-1}$$

where α is a smoothing factor $0 < \alpha < 1$, x_t is the raw data point at time t and s_t is the smoothed data point at time t for $t > 0$

For smoothing the location counts in our case, one may select a few different geographical orderings of locations (e.g., east to west, west to east, etc.) and propagate the weights in the ordering direction. Suppose $l_e^s = \{(l_1, c_1), (l_2, c_2), \ldots, (l_n, c_n)\}$ is one such ordering. Then updating c_i with

$$\delta c_i + (1 - \delta) c_{i-1} \tag{4.1}$$

for $\delta \in [0, 1]$ and $i = 2, \ldots, n$ will give a smoothed signature. In our experiments, we do this smoothing for siblings under two orderings (east to west and west to east) and with δ set to 0.6 based on cross-validation.

A mismatch between locations can also happen if they are reported at different levels of dispersion. For example, the headline

```
Floyd was killed in police custody
```

is reported under the location *United States* in global news, whereas it is reported under *Minneapolis* in the local news. To handle inference between different levels of location dispersion, we may transfer weights from a lower level of dispersion to an upper levels while constructing the signature. This can be done in our smoothing by updating the weight of a parent l_i based on the sum of the weights of its children in the signature vector, i.e.

$$c_i = \delta c_i + (1 - \delta) \sum_{l_j \in children(l_i)} c_j \tag{4.2}$$

where $\delta \in [0, 1]$. In our experiments, δ here is set to 0.5 based on cross-validation.

4.5 Spatial signature of mentions

A location signature can be constructed for each mention based on the location cues in the surrounding text as well as in the metadata description. Under the inheritance hypothesis [56], named entities inherit the location of an article (e.g. the location where a headline is published or a tweet is posted). The location signature can also be smoothed following the same algorithm discussed for smoothing entity signatures (equations 4.1 and 4.2). With the signatures of entities and mentions described as vectors, a spatial similarity is be computed using cosine or the inner product between the vectors.

Figure 4.1 shows the steps for constructing a signature for a candidate entity. Given an entity Wikipedia page, we tag all locations in the page using a NER tool. The frequency of each location is the number of times it is tagged as location in the page. We search each location in Geonames and apply location disambiguation as discussed in the section 4.3. Our initial embedding will have information about all places tagged in the entity page at the city level as well as its neighbours and parents at the level of province/state and country. As Edmonton is a city (Feature class 'A'), its neighbours within a radius of 60km include Beaumont, Bon Accord, Calmar, Leduc, Nisku, St.Albert, Spruce Grove and Strathcona County. With exponential smoothing, some

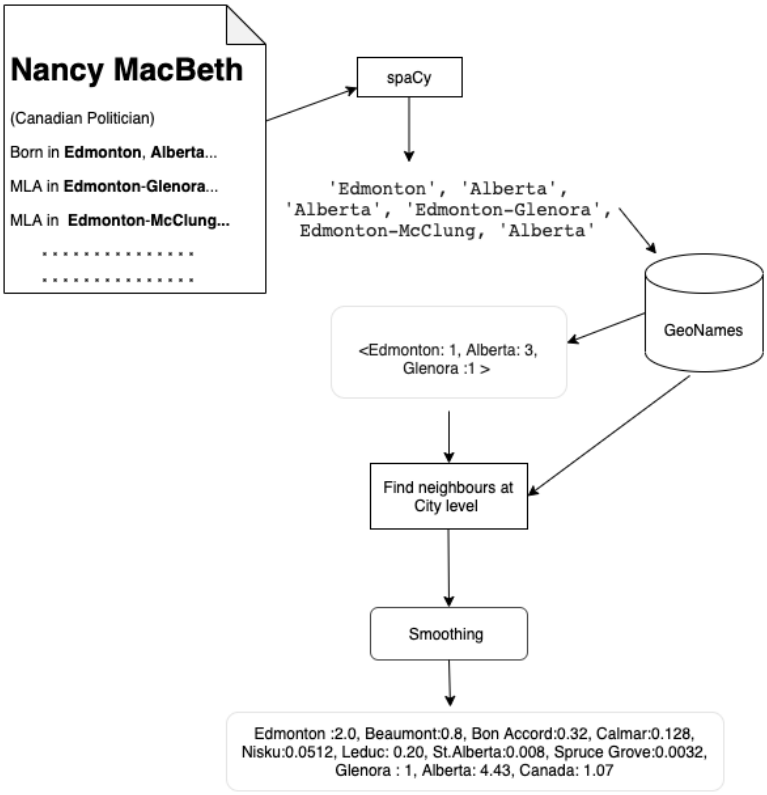

Figure 4.1: Construction of Spatial Embedding for mentions and candidate entities

weights are transferred from Edmonton to its neighbours and its parents Alberta and Canada, based on Equation 4.2.

Chapter 5

Probabilistic Location Aware NED Framework

This chapter presents a probabilistic local NED framework that combines local features with spatial signals to enhance the disambiguation process. The framework combines the prior probability of an entity being mentioned, the similarity between the contexts of the mention and a candidate entity, and the spatial similarity between the mention and a candidate entity.

5.1 Tagging mentions

We consider an input text with mentions of named entities in ambiguous surface forms and aim to link each mention to a proper entry in the knowledge base, thus giving each mention in text a disambiguated meaning. Examples of input text include a tweet, a news headline and a search query. This input text is tagged, and the mentions of named entities are detected using standard tools (e.g., the Stanford NER tagger [35]), which can be of type person, organization, location, event, song, etc. We assume the input has some location cues in the form of either locations or some geo-coordinates for our approach to be applicable.

5.2 Finding candidates

Public knowledge bases (e.g. Yago and DBpedia) provide a good collection of candidates, where each entity has a short name (e.g., *"Apple"* for Apple Inc)

and a set of paraphrases (e.g., *"Big Apple"* for New York City), constructed from Wikipedia disambiguation pages, redirect pages, and anchor links. For news headlines, the names are expected to be accurate, and one may select all entities where either the title or a paraphrase matches the mention fully. A full match will not work for tweets and search queries, which generally have misspellings, abbreviations, and unreliable capitalization. For such input, candidates may be selected using a k-gram matching approach [60]. The names match if the k-gram similarity between the mention and the canonical name of a candidate entity is more than a specified similarity score. In our experiments with tweets, k was set to 3. We use Yago's repository of entities as our candidate set (as done in other works [51]).

5.3 Prior popularity

The prior probability is a context-independent feature that denotes the prominence or popularity of an entity in the candidate set of entities given for a mention. It is usually estimated from the Wikipedia-based count information for each name mentioned in the anchor texts of links referring to specific entities, i.e. number of inlinks [43], [51], [61]. Very few authors use Wikipedia view statistics and both inlink and outlink count information [21], [41]. The idea is for each anchor text that contains a name, how often it refers to a particular entity. This provides for each name a probability distribution over candidate entities. For example, "Obama" refers to "Barack Obama" in 60.5% of the occurrences and to "Obama, Fukui (location)" in 2.4% of the cases.

5.4 Context similarity

Various forms of context similarity of a mention and a candidate entity have been utilized in the literature. The context of a mention can be the bag of words collected from the entire input text [19], [41] or a suitable window around the mention in the document [47], [61]. The context vector of a candidate entity may be composed of keyphrases [51], anchor text [61], categories [19], [21] and Wikipedia Info boxes [21]. Every entity in our KB is associated with a

textual description in the form of keyphrases which can be compared to the surrounding context of the mention. The more the context and the entity description overlap, the stronger the signal for a correct entity. We use the keyphrase-based similarity of Hoffart et al. [51], which measures the mutual information between the keyphrases of an entity and the words that appear within the context window of a mention. For short text such as news headlines and tweets with less context, one can use all tokens in the entire input text (maybe excluding the stopword and mention itself) as context on the mention side. On the entity side, keyphrases are extracted from an authoritative source for each entity, for example the homepage of an organisation or an individual or a Wikipedia article. Here we use the keyphrases extracted from the Wikipedia article of the entity, the anchor texts of links to the article, category names, titles of article linking to the entity article, citation titles, etc. These keyphrases are expected to appear in the context window of the mentions of the entity. Note that the keyphrase is multi-word; each word in the keyphrase which is a keyword is a single-word. To calculate the keyphrase-based context similarity, we first need to find the notion of how important the keyphrase is to a given entity. For this purpose, we calculate the Normalized Pointwise Mutual Information (NPMI) to quantify the likelihood of co-occurrence of two words (entity and keyword), considering the fact that the frequency of the single word may increase the likelihood.

Let $KP(e)$ denote the keyphrase set of an entity e. For each word w that occurs in a keyphrase, NPMI between the entity e and the keyword w is calculated as follows,

$$NPMI(e, w) = \frac{PMI(e, w)}{-\log p(e, w)}, \tag{5.1}$$

where

$$PMI(e, w) = log\frac{p(e, w)}{p(e)p(w)} \tag{5.2}$$

keywords with NPMI(e,w) ≤ 0 are discarded for the NED as $\log p(e, w)$ becomes $-\infty$ indicating that there is no co-occurrence of entity and the keyword.

The co-occurrence probability $p(e, w)$ of an entity e and keyword w is given by the number of times w appears in the union of its keyphrases with the keyphrases of all entities linking to it. The co-occurrence probability $p(e, w)$ is given by,

$$p(e, w) = \frac{|w \in (KP(e) \cup KP(e^{'})|}{N} \tag{5.3}$$

where $e^{'} = IN(e)$ is set of all entities that link to e and N denotes the total number of entities.

It is not possible to see the exact keyphrases of an entity in the context of input text. Hence, it is necessary to compute the similarity of the mention m and the entity e, taking into account the partial match of e's keyphrases in the text. Consider an input text mentioning "Obama" with the context "United States president". Their corresponding candidate entity "Barack Obama" has one of its keyphrase as "44th president of United States" where there is a partial match between the context of m and e. For each keyphrase, we usually find the shortest window of words that contains a maximal number of words in the keyphrase. As we use short text, we consider the window to be all the tokens from the input. The score of partially matching phrase q is given by

$$score(q) = Z\left(\frac{\sum_{w \in window} weight(w)}{\sum_{w \in q} weight(w)}\right)$$

where,

$$Z = \frac{\#\text{ matching word}}{\text{length of window(q)}}$$

and $weight(w)$ is NPMI weight.

The context similarity score is given by aggregating over all keyphrases of e and their partial matches in the input text.

$$cxtSim(m, e) = \sum_{q \in KP(e)} score(q)$$

5.5 Spatial similarity

Spatial vectors for each entity and mention is constructed as described in Section 4.4 and 4.5 of Chapter 3. The similarity between the feature vector of an entity and that of a mention may be measured using cosine or the inner product.

5.6 Objective function

Given an input text with a set of mentions $\{m_1, m_2, \ldots, m_n\}$ and a candidate set of entities from a knowledge base, we want to map each mention to either a unique entity in our candidate set or $\emptyset$ for out-of-KB entities. Our framework combines prior probability, context-similarity and spatial similarity into an objective function, and we want to find an assignment of candidates to mentions that maximizes the confidence score. Let *cad(m_i)* denote the set of candidates of mention m_i. Our objective function can be written with the goal of finding:

$$e^* = \underset{e_{j,i} \in cad(m_i)}{\operatorname{argmax}} \Bigg(\alpha . \sum_{i=1}^{n} prior(m_i, e_{j,i}) + \beta . \sum_{i=1}^{n} cxtSim(m_i, e_{j,i}) + \gamma . \sum_{i=1}^{n} locSim(m_i, e_{j,i}) \Bigg) \tag{5.4}$$

where $\alpha + \beta + \gamma = 1$, and $prior()$, $cxtSim()$ and $locSim()$ respectively denote the prior probability, the context similarity and the spatial similarity of a mention and a candidate entity. Illustration about the framework is shown in Figure 5.1.

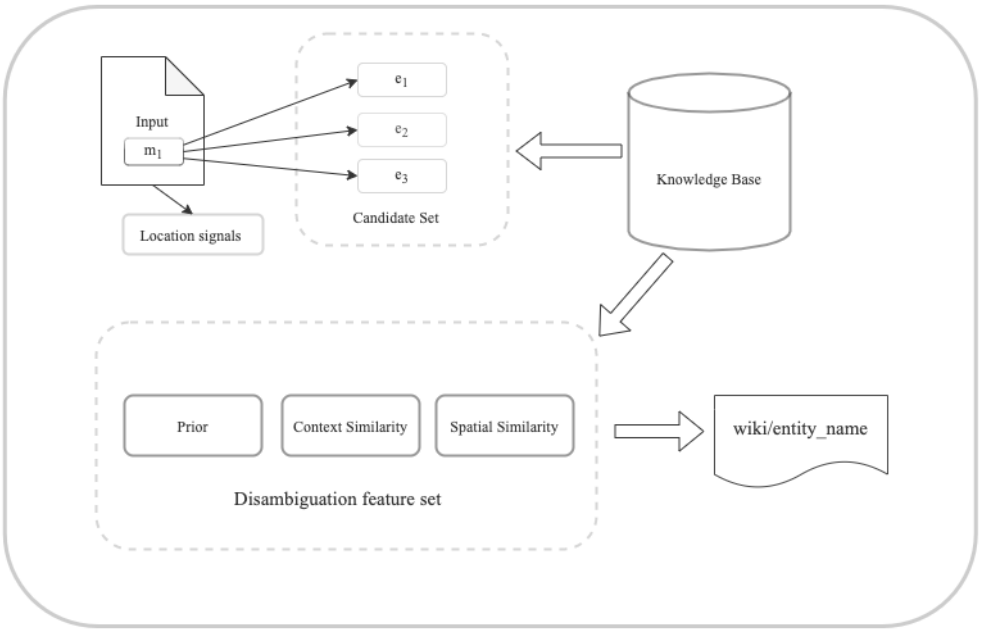

Figure 5.1: Probabilistic Location-Aware NED Framework

Chapter 6

Neural Location Aware NED Framework

This chapter presents an end-to-end neural EL framework that jointly discovers and links entities in text unlike the previous framework that only addresses the entity disambiguation task without leveraging mutual dependency. In this model, we explore the semantic relation between the local context and a candidate entity, captured via a distributed representation of words and combine them with spatial and prior feature to disambiguate mentions in a short text. The key components of this framework are character embeddings, word embeddings, entity embeddings, spatial embeddings and prior probability.

6.1 Task of Interest

Given a short text (a query or tweet) as a sequence $D = w_1, w_2, ..., w_k$ of words $w_i \in \mathcal{W}$ from a dictionary along with some location cues. The output of the model is a list of mention-entity pairs $\{(m_i, e_{ij})\}_{i=1,2,...,n}$ where each mention m_i is a word subsequence of input text $m = w_q, ..., w_r$ and each entity $\{e_{ij}\}_{j=1,2,...,t}$ is an entry in a KB (eg Wikipedia). For training, we do not require expensive manually annotated negative examples and the input comes with a gold mention - entity pair $\mathcal{G} = \{(m_i, e_i^*)\}$

The following sections will describe the components of our neural end-to-end EL model as depicted in Figure 6.1. $\log p(e_j|m)$ denotes the prior probability which is described previously in Section 5.3 in Chapter 4. The

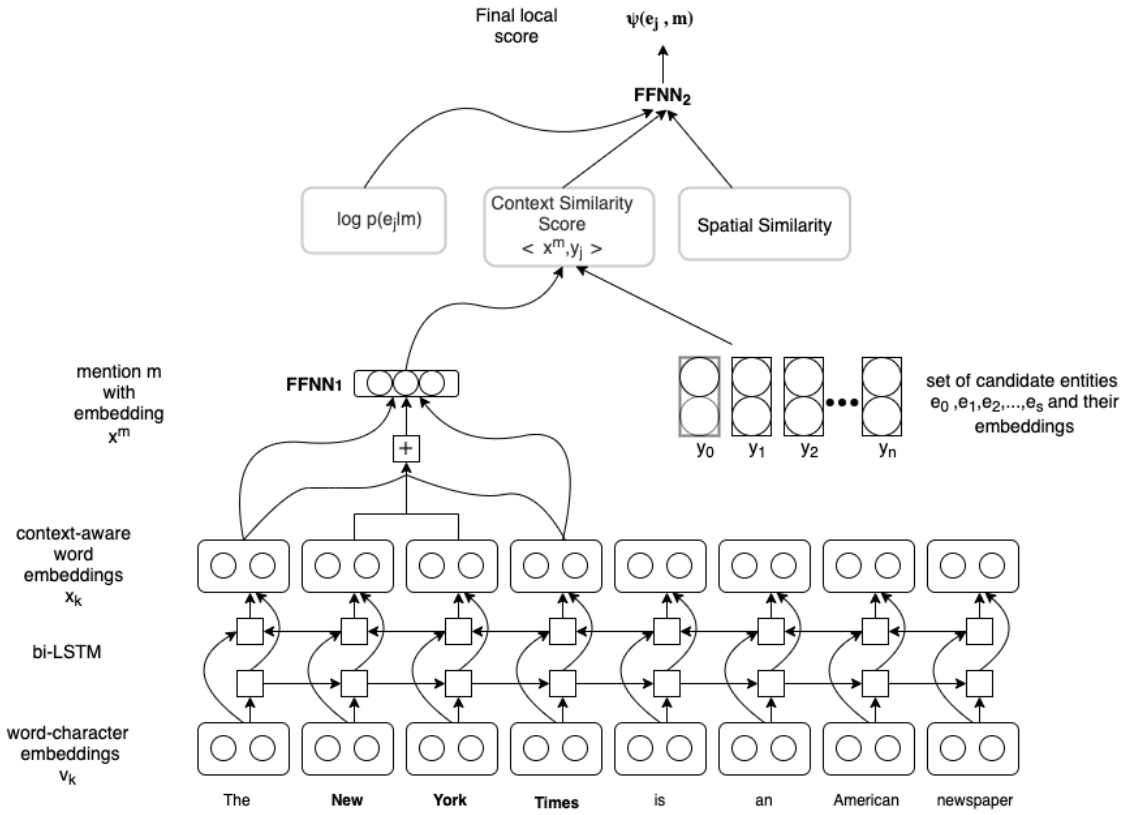

Figure 6.1: Our local neural model with all the features from the framework. The final score is used for both the mention decision and entity disambiguation

context similarity score is calculated between context-aware word embedding and entity embedding. We also add spatial similarity in the feature set and use two Feed Forward Neural Networks (FFNN) for projecting the output in this framework.

6.2 Local context feature

To disambiguate a mention to its correct entity, it is crucial to capture the information from its context. We construct a dense contextualized vector representation from word and character embeddings for this purpose by inducing it with information about surrounding words.

6.2.1 Word Embeddings

Short texts are very noisy and sparse in the use of vocabulary. In traditional textual representation such as tf-idf, a word is represented by a one-hot vector, i.e. the vector consist of 1s in the cell representing the word and 0s in the rest of the cells. A word's term frequency (tf) gives the number of times a word occurs in the considered document, and word's document frequency (df) is the number of documents in the corpus that contains that specific word. The inverse document frequency (idf) measures the informativeness of terms by diminishing the weight of terms that frequently occur in the document set and increasing the weight of terms that occur rarely. A tf-idf based similarity measure is based on exact word overlap. As the text becomes smaller in length, the probability of having words in common decreases.

Further, these count-based measures ignore the synonyms and any semantic relatedness between different words and are prone to the negative effect of homonyms. Instead of relying on word overlap, we explore the option of word representation incorporating semantic information into the similarity process. We use word embeddings as a building block to construct text representation. These embeddings are distributed vector representations of a single word in a fixed-dimensional semantic space. These distributed representations of words learned by neural networks have shown significant performance improvement compared to the well-known Latent Semantic Analysis and Latent Dirichlet Allocation for preserving linear regularities among words [74], [123]. Word embeddings have been used to help achieve better performance in several NLP tasks [18]. We use word2vec by Mikolov et al. [74] that utilizes two algorithms to produce computing continuous vector representations of words from very large data sets: Continuous bag-of-words (CBOW) and Continous Skip-gram model. Figure 6.2 shows CBOW and Skip-gram model architecture. The training objective is to learn a word vector representations of words that are good at predicting the nearby words. CBOW learns an embedding by predicting the current word based on its context, and the continuous skip-gram model learns by predicting the surrounding words given the present word.

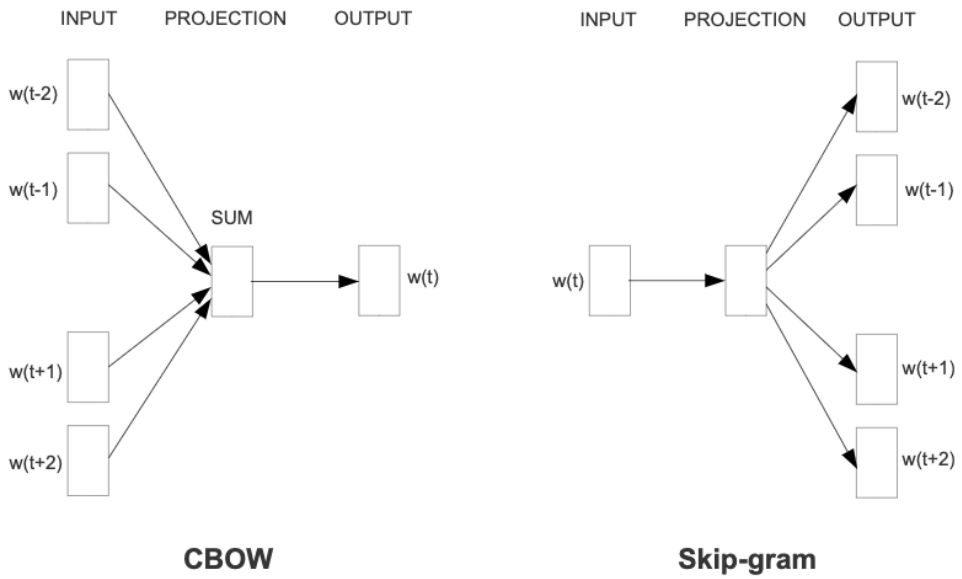

Figure 6.2: CBOW and Skip-gram model taken from Mikolov et al. [74]

These models learn about the word from their usage contexts, and the context window size sets the coverage of neighbouring words. These word representations computed using neural networks are fascinating because the learned vectors encode linguistic regularities and patterns. Many of these patterns can be represented as linear translations. For example, by performing simple algebraic operations on the word vectors we see that the vector("Madrid") - vector("Spain") + vector("France") is closer to vector("Paris").

In this framework, we use pre-trained word2vec vectors. There are other word embeddings like Glove developed by Pennington et al. [87] and contextual embeddings like ELMo [91] and BERT [20]. However, a word2vec based EL provides competitive performance compared to EL based on other word embeddings like BERT [100].

6.2.2 Character based representation of words

The input layer to our model is the vector representation of individual words. However, these word embeddings are not just enough to represent the words in the input document. Many languages have orthographic or morphological evidence that something is a name (or not a name); we want representations sensitive to the spelling of words. Although these pre-trained word embeddings are trained with large amounts of data to learn semantic and syntactic similarities between words, each vector is independent. For example, English speakers can understand that *dog* and *dogs* are closely related and that they are only differentiated by plurality morpheme "-s" bound to noun phrases. Though word embeddings are known to learn that cats, kings and queens exist in roughly the same linear agreement as to the *cat, queen, and king* do, the model does not represent that adding an -s at the end of the word is an evidence for this transformation. This, in turn, means that the word lookup table cannot represent unseen words like *Frenchification* even if the components *French* and *-ifications* are observed in another part of the text [67].

We, therefore, use a model that constructs representations of words from the representations of the characters. Learning a word representation from characters comes with its advantages. First, we can learn character-level features without needing hand-engineered prefix and suffix information about the word. Second, learning character-level embeddings helps in learning representations that are specific to the task and the domain at hand. Character embeddings have been found helpful for morphologically rich languages and to handle out of vocabulary problems for tasks like part-of-speech tagging, language modelling and dependency parsing [6], [62], [67].

RNNs and LSTMs [49] are capable of encoding very long sequences. However, they only look at the most recent information/input. As a result, we expect the final representation of a forward LSTM to be an accurate representation of the suffix of the word and the final state of the backward LSTM to be a better representation of its prefix. Alternatively, CNNs have been proposed for the representation of words from their characters [58], [120]. However, CNNs

are designed to discover position-invariant features of their inputs. While they are appropriate for image recognition (a cat can appear anywhere in a picture), the critical information in our work is position-dependent (prefix and suffix). Hence LSTM is a better function class for modelling these representations in our work.

6.2.3 Context Independent Word Embedding

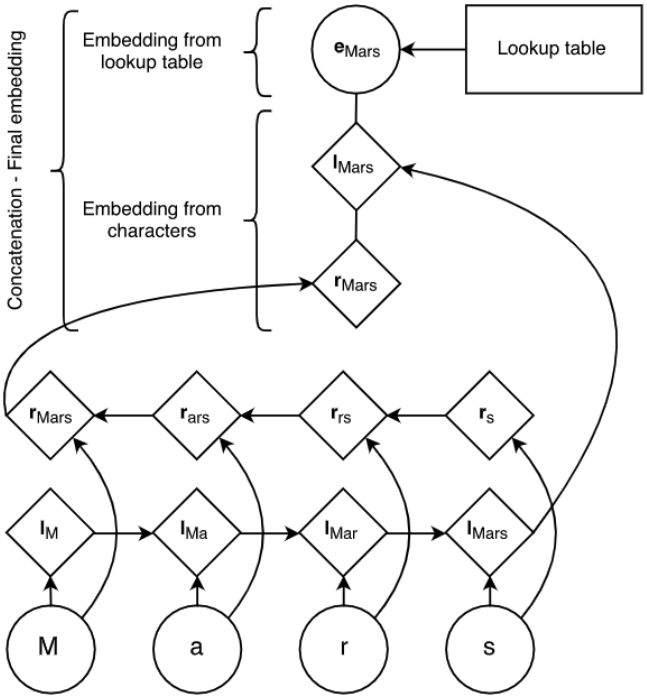

Figure 6.3: The Character embeddings of the word "Mars" are given to bidirectional LSTMs. We concatenate their last output to word embeddings from word2vec. The figure is taken from lample et al. [62]

A Character lookup table is initialized at random and it contains an embedding for every character. We train a character embedding in addition to the word embedding using bi-directional LSTM [59], [62]. The character embeddings, corresponding to every character in the word are given in direct and reverse orders to a forward and a backward LSTM. Let $\{z_1, ..., z_L\}$ be the character vectors of word w. The forward and backward LSTM can be defined

recursively as

$$h_t^f = FWD - LSTM(h_t - 1^f, z_t),$$

$$h_t^b = BKWD - LSTM(h_t + 1^b, z_t)$$

The character embedding of w is the concatenation of the hidden states of the forward and backward LSTM $[h_L^f, h_1^b]$ where h_L^f is the hidden state of the last character of forward LSTM and h_1^b is the hidden state corresponding to the first character of the backward LSTM. This character-level representation is then concatenated with a word-level representation from the word-lookup table (in our case pre-trained word2vec) to form a context-independent word-character embedding, i.e. $\{v_k\}_k \in 1, ..., n$. This word-character embedding will prevent the models from being dependent on one representation or the other too strongly. Figure 6.3 shows character embedding trained using bi-LSTMs for the word "Mars". l_{Mars} denotes the representation of the character from left to right, and r_{Mars} denotes the representation of characters in the reverse order. The final output from the forward and backward LSTMs are concatenated with the corresponding word embeddings of the word "Mars."

6.2.4 Context Dependent Word Embedding

We find it essential to make word embeddings aware of their local context. Inducing the context information may help in the mention boundary detection and entity disambiguation tasks by leveraging the contextual cues. We use bi-LSTMs on top of the word-char embedding $\{v_k\}$ to encode the context information into the words. The hidden states of the forward and the backward LSTMs corresponding to each word is then concatenated into a context-aware word embedding. This context-aware word embedding sequence is denoted as $\{x_k\}_k \in 1, ..., n$. The hidden states represent the left and the right context of the word. A context-aware word embedding will prevent the embeddings from having the same representation for the word type regardless of the context of the word token. This model is similar to the line of work by kolitsas et al., which represents an entity mention as a combination of LSTM hidden states [59] and that of Gupta et al. which concatenate outputs of two

LSTM networks that independently encode left and right contexts of a mention(including the mention itself) [44].

6.2.5 Self-Attention Mechanism

Syntactic heads are typically included as features in previous systems [22], [63]. For each possible mention, we produce a fixed size representation learning from the notion of headness[1] using an attention mechanism [5] over words in each span. Given a mention $m = w_q, \ldots, w_r$, we concatenate the embeddings of the first,last and soft head word of the mention:

$$g^m = [x_q; x_r; \hat{x}^m]$$

$$\alpha_k =< w_\alpha, x_k >$$
$$a_k^m = \frac{exp(\alpha_k)}{\sum_{t=q}^{r} exp(\alpha_k)}$$
$$\hat{x}^m = \sum_{k=q}^{r} a_k^m . v_k$$

where $\hat{x}^m$ is the weighted sum of word vectors between word span w_q and w_r known as head word vector. The weights a_k^m are automatically learned and correlate strongly with traditional definitions of head words. The soft head embedding in our case marginally improves the results, probably due to the fact that most mentions are at most two words long.

We then project g^m into a shallow FFNN to get the mention embeddings x^m, which are of the same size as the entity embedding and learn non-linear interaction between the word vectors of the mention.

$$x^m = FFNN_1(g^m)$$

6.3 Entity Embeddings

Entity embedding provides a semantic representation of entities in low-dimensional space. They are bootstrapped from word embeddings and are trained independently for each entity. The line of work on word embeddings is extended

[1] In the Transformer, the Attention module repeats its computations multiple times in parallel. Each of these is called Attention Head.

to entities for disambiguation tasks [25], [53], [115], [125]. We use the pre-trained entity embedding of Ganea et al. [38]. The authors bootstrap each entity embedding from the canonical entity page, i.e. KB description of the entity and the local contexts of the hyperlinks to the entity page instead of leveraging entity-entity co-occurrence statistics which suffer from sparsity and large memory usage [115].

A generative model is used where the co-occurrence of word w with an entity e is sampled, which is the approximation of word-entity conditional distribution $\hat{p}(w|e)$. An exponential model is trained to approximate the word-entity conditional distribution $\hat{p}(w|e)$ given as,

$$\left(\frac{\exp(\langle x_w, y_e\rangle)}{\sum_{w' \in W} \exp(\langle x_{w'}, y_e\rangle)}\right) \simeq \hat{p}(w|e)$$

where x_w is pre-trained word vector and y_e is the entity embedding that is needed to be trained.

–

6.4 Candidate Generation

An essential part of EL is candidate generation. A brief discussion of candidate generation is already given in Chapter **5** and in Section **5.2**. Candidates are generated using three methods: (1) based on a surface form matching, (2) based on an expansion using aliases and (3) based on a prior probability computation. It is common to apply multiple approaches to candidate generation. The resource constructed by Ganea and Hofmann [38] relies on the prior probabilities obtained from entity hyperlink count statistics from CrossWikis [106], a cross-lingual dictionary for Wikipedia concepts and Wikipedia as well as on the entity aliases obtained from "means" relationship of the YAGO [51] dictionaries. We denote the candidate set as $C(m)$ and use it during the training and test. We select top-k candidates (k=15) in such a way that the top 5 entities are based on mention-entity prior probability [106] $p(e_{ij}|m_i)$ and top 10 candidates are based on the context-entity similarity computed by constructing a context vector and simply averaging all its corresponding word vectors

for this purpose. Choosing top-k candidates is for optimizing the memory and the run time.

6.5 Spatial Embeddings

Spatial vectors for each entity and mention is constructed as described in Chapter 3. The similarity between the feature vector of an entity and that of a mention may be measured using cosine or the inner product and is denoted as locSim$(m_i, e_{i,j})$

6.6 Local Score for Disambiguation

For each mention m_i that has more than one candidate entity, i.e., $\mid C(m_i) \mid \geqslant 2$ and for each candidate $e_{i,j} \in C(m)$, we compute both the similarity score between the context-aware mention embedding x^m and the entity-embedding y_j and the similarity score between the spatial embedding of mention m_i and entity $e_{i,j}$ to capture useful information for mention detection and ED tasks. We then combine the context similarity, spatial similarity and the log prior probability using shallow FFNN to get a total local mention-entity score,i.e.

$$\Psi(m_i, e_{i,j}) = FFNN_2([logp(e_{i,j}|m_i); < x^m, y_j >; locSim(m_i, e_{i,j})])$$

Chapter 7

Results and Discussions

7.1 Experimental Evaluation

This section presents an experimental evaluation of our algorithm. We review our dataset and preprocessing, describe our experimental setup and present an evaluation of our algorithm, in comparison with different baselines from the literature on Probabilistic and Neural Framework, on two datasets: a NYT-headlines dataset and a geotagged Tweets dataset.

7.1.1 Datasets

The standard dataset for NED evaluation (e.g., MSNBC news-wire articles [19], CoNLL-YAGO [51], TAC KBP [55], and ACE 2004 NED [95], AQUAINT [75]) are high-quality news articles with most entities mentioned at least once by their full names. To gauge the effectiveness of our spatial signatures, we evaluate our named entity disambiguation on short text, with both formal and informal structure. We use a subset of the New York Times archive (originally containing 1.8 million articles published between 1987 to 2007) [98], extracting only headlines that have no more than two mentions per headline. This results in 2340 headlines. To validate our model in informal text, we use tweets with two classes of geographical metadata. In the first class, the tweet location coordinate is given as latitudes and longitudes. For the second class, the tweet location is set to the home location of the users as set in the account. In our model, we use the location cues both from the context and geotag with coordinates. The dataset we use to evaluate our framework does not come with

home location geo coordinates. Hence we do not use them with other location cues.

The Locke collection [68], Habib collection [46] and Micro post-collection [14] are all re-annotated by Habib et al. [45] with a total of 5535 named entities in tweets linked to Wikipedia. Less than 2% of those tweets have any location context in the tweet for us to use in our evaluation. Hence, we use a subset of a tweet dataset from Farazi and Rafiei [28] that originally contained 53 million geo-tagged tweets, mostly from the US and Canada. We randomly took only 3490 tweets that had at least one tagged named entity mention for preprocessing and added 314 tweets from a dataset by Fang and Chang [26]. We cleaned the tweets by removing hashtag symbols (#), retweets (RT), @ symbols, and did a text segmentation using the ekphrasis[1] library.

We used the Stanford NER on our tweets dataset, after evaluating its performance with other NER Tools including spaCy, NLTK[2], and TwitterNER [77]. The Stanford NER had a precision around 0.67 on 200 tweets chosen randomly comparing to spaCy, NLTK and TwitterNER which had precision' around 0.62, 0.58 and 0.65 respectively. We hand-annotated 1762 gold entities from the NYT subset and 1015 entities from the Tweet subset, with corresponding entities in the Wikipedia 2020 dump. We removed NER errors, out of KB entities and entities with full names in the input text. To avoid any discrepancy between different Wikipedia versions that were used as the source for our features (Priors and Keyphrases in AIDA were collected from Wikipedia 2014 and our spatial signatures were constructed from Wikipedia 2020), we annotated only those entities that existed in both years.

7.1.2 Evaluation Metrics

We use standard accuracy, precision, recall, and the F1 measure as evaluation metrics. A dataset may contain many single documents to have a micro or macro measure of precision, recall and F1 measure. The micro measure shows the performance over all annotations inside the dataset, and the macro measure

[1] https://github.com/cbaziotis/ekphrasis
[2] www.nltk.org

shows the average performance per document. Since we have short text with at most two mentions per document with only one mention per document most of the time, we calculate precision, recall, F1 and accuracy aggregated across mentions (micro-averaged):

$$accuracy = \frac{|true \cap pred|}{|true \cup pred|}, \tag{7.1}$$

$$precision = \frac{|true \cap pred|}{|pred|}, \tag{7.2}$$

$$recall = \frac{|true \cap pred|}{|true|}, \tag{7.3}$$

$$F1 = 2.\frac{precision.recall}{precision + recall} \tag{7.4}$$

where, *true* is the ground truth entity and *pred* is the linked entity by the NED systems.

We use spatial feature with the two frameworks explained in **Chapter** 4 and **Chapter** 5. We will see each of them in the sections below.

7.1.3 Evaluation of the Probabilistic Framework

We evaluated our model against the feature set of AIDA, as we added the spatial dimension to the feature set and retrained the model to get new weights for the features. The weights for prior, context and spatial feature were $\alpha = 0.21$, $\beta = 0.33$, $\gamma = 0.46$ for the NYT headlines and $\alpha = 0.5$, $\beta = 0.2$, $\gamma = 0.3$ for the tweet subset based on cross validation. We chose the α, β and γ values that gave top accuracy on different trials. This feature weighting highlights the importance of each feature class for the dataset. The tweet subset had more prominent entities hence larger prior weights, whereas the NYT subset had more varied locations around the globe and a larger weight to spatial features. From Table 7.1, we combined the spatial feature with prior probability and were able to see around 14% increase in the accuracy (rows 1 and 2) for NYT dataset. We then combined the spatial feature with both the prior and the

context and was able to see around 8% increase in the accuracy. However we only saw 3-4% increase in the overall accuracy of the framework with the tweet subset. Overall we see a significant rise in the accuracy when combining the prior and the context feature with a spatial dimension for both datasets.

Feature set	NYT subset	Tweet subset
prior	45.7	33.2
+spatial	**59.1**	**36.4**
prior+context	65.4	40.4
+spatial	**73.8**	**44.9**

Table 7.1: Micro-accuracy of our model with and without spatial feature for probabilistic NED

NED system	NYT subset	Tweet subset
xLisa [119]	54.1	35.8
AGDISTIS [110]	58.4	33.7
WAT [32]	52.9	33.2
TagMe 2 [33]	60.2	43.7
Babelfy [79]	63.1	41.2
KEA [112]	57.9	40.8
AIDA [51]	65.7	42.6
WNED [41]	72.1	41.4
reimpl. diaNED-1[1]	68.9	-
Probabilistic Framework	**74.5**	**45.6**

Table 7.2: Micro-F1 of various NED systems on NYT-subset and Tweet-subset

Table 7.2 compares the micro F1 of our model with that of different NED systems available via GERBIL [111] and to our re-implemented diaNED [1]. Many of the tweets in our tweets dataset did not have a temporal metadata, hence we only evaluated diaNED on the NYT subset using the year of publication as the temporal context. Though we could not directly compare our model with the systems from GERBIL as they differed in the candidate generation and entity selection process, cynically, the table shows that our model enhances the NED quality by adding the spatial signatures. Table 7.2 also

Entity Type	NYT-Dataset		Tweet-Dataset	
	w location	w/o location	w location	w/o location
PER	40.6	27.9	43.7	17.1
ORG	36.9	18.3	24.9	18.8
LOC	15.2	14.7	25.3	22.8

Table 7.3: Micro-accuracy of our model on NYT subset and Tweet-subset with and without spatial awareness.

shows that incorporating spatial signature as a local feature gives a performance comparable to that of models with global features.

To further study the importance of spatial signals, we probe the results of our model in the NYT subset and Tweet subset to see how location-awareness helps improve the disambiguation quality based on different entity types. From Table 7.3, we can see that the location feature helps entity types person and organization with the spatial signals bounding these entities. However, we can see a very moderate improvement for the location type using the spatial feature. To better understand why location entity-type has less improvement with spatial signals, we examined all the location entities from NYT and (which constituted 15% of total mentions) and noticed that nearly 75% of the disambiguation answers came from the prior probability and context. There were not enough locations in the context to see more improvement for the location entity type. The tweet subset had overall 7% of total mentions as locations in the context out of which more than 70% disambiguation decision came from prior probability. Here are two examples from our tweet subset that shows how the location signal helps linking to correct entities in Wikipedia.

```
(T1) Bro was my favorite player after Kobe..(USA)
(T2) Drake My Go To For Everyyy Mood (Texas, USA)
```

In *T1*, for the mention *Kobe*, the cue *USA* helps to differentiate between the candidate *Kobe, a city in Japan*, and the gold candidate *Kobe Bryant*. Similarly, In *T2*, the location cue *Texas, USA* helps to differentiate between Ervin Drake, American song writer, and Drake (Rapper); the latter has a stronger tie to Texas and is the correct entity in this case.

7.1.4 Evaluation of the Neural Framework

There is not a standard short documents benchmark for ED and we need location context for the training set. We train our model on the biggest publicly available EL dataset, AIDA/CoNLL, consisting of a training set of 18,448 linked mentions in 946 documents, a validation set of 4791 mentions in 216 documents and a test set of 4485 mentions in 231 documents. We have two training settings: (TR-I) entire news article for training,
(TR-II) Only headlines for training
For TR-I, We use 900 documents from the AIDA-training dataset with location either in meta-data or context. For TR-II, we take all the headlines from the CONLL-AIDA dataset that have a location in the meta-data and context, resulting in 900 headlines. We use the NYT corpus and Tweet subset as both the validation and the test set.

Our pre-trained word and entity embeddings are 300 dimensional vectors, whereas our character embedding is 50 dimensional vector. Character LSTMs have hidden dimensions of 50. Thus word-character embeddings are 400 dimensional vector. The contextual LSTMs have a hidden size of 150, resulting in 300-dimensional context-aware word vectors. Dropout is added in the concatenated word-character embeddings on the output of the bidirectional context LSTM and the entity embeddings in FFNN2. The two FFNNs in our model are simple projections without hidden layers. We use atmost 25 entity candidates per mention both at train and test time. For the loss optimization, we use Adam with a learning rate of 0.001 and perform an early stopping by evaluating the model on the validation set every 10 minutes and stop after ten consecutive evaluations with no significant improvement in the micro F1 score.

We train two neural models denoted as "base model" and "base model + att". The base model only includes the prior probability and context similarity feature, and the base model+att includes a base model plus a head attention mechanism. We also tried using long-range context attention from Kolitsas et al. [59] in our feature set but adding that feature did not improve our results as

our dataset was short text. Hence, we did not use long-range context attention for subsequent analysis. We use our spatial feature with the base model and base+att model and report the scores.

Table 7.4 shows the result of adding the spatial feature to log-prior, neural context and head attention mechanism while training on the full news corpus. Table 7.5 shows the result of adding the spatial feature to log-prior, neural

Feature set	NYT subset	Tweet subset
Base Model	84.2	59.7
+spatial	**85.1**	**62.4**
Base Model+att	83.3	59.2
+spatial	84.2	60.7

Table 7.4: Micro-F1 of our model with and without spatial feature using TR-1 as training

Feature set	NYT subset	Tweet subset
Base Model	77.4	53.2
+spatial	**82.3**	**56.8**
Base Model+att	77.2	52.1
+spatial	79.8	54.5

Table 7.5: Micro-F1 of our model with and without spatial feature using TR-11 as training

NED system	NYT subset	Tweet subset
Falcon [97]	75.2	47.5
K-NED [31]	70.3	51.8
Base+Spatial	82.3	56.8

Table 7.6: Micro-F1 of our model with other EL models for short text

context and head attention mechanism while training on TR-II. The head attention mechanism is used when a mention has more than two or three tokens (e.g. New York Times). Though the training set on TR-1 has some mentions three or more tokens long, NYT and Tweet subsets have mentions at most two words long. This might have decreased the performance of the base model with attention mechanism (denoted as Base+att model) compared to

Entity Type	NYT-Dataset		Tweet-Dataset	
	w location	w/o location	w location	w/o location
PER	49.6	36.9	33.6	21.1
ORG	36.9	27.5	40.4	16.2
LOC	10.8	10.3	30.1	29.5

Table 7.7: Micro-accuracy of our model on NYT subset and Tweet-subset with and without spatial awareness.

the base model in Tables 7.4 and 7.5.

We evaluate our base + spatial model against Falcon [97] and K-NED [31] whose codes are publicly available. For training the neural model of K-NED, we used TR-II as training and NYT subset and Tweet subsets for testing.

Similar to our analysis of the probabilistic model, we also probe the results of our model in the NYT-subset and Tweet subset on different entity types. From Table 7.7, we can see that the location feature helps entity types person and organization with the spatial signals bounding these entities. However, we can see a moderate improvement for the location type using the spatial feature even in the neural model. This may be because more than 75% of the location in the test set, i.e., NYT subset and tweet subset, are learned mainly from the training set.

Chapter 8

Conclusion and Future direction

This thesis proposes a NED approach that explicitly considers the location to aid the disambiguation process. We offer an unsupervised framework to identify and create a spatial signature for the mentions in an input text and entities from a knowledge base. We then utilize them to enhance the disambiguation quality of short text in formal and informal settings. Our evaluation results on both the probabilistic and neural framework show that location-awareness improves the NED quality when the entities in the text hold some regional boundaries. There are a few potential directions for improving this framework in the future. First, our work may be extended to have a more robust resolution of entities with global boundaries (e.g., Justin Bieber, who tours around the globe). Second, our work may be extended to support inference on missing locations based on other metadata present in the text. Third, our analysis of how location helps disambiguate specific named entities was only done at a coarse level which includes PER, LOC and ORG. In the future, we want to analyze this for more fine-grained entity types like books, events, arts, etc. Lastly, we would like to experiment using contextual word embedding such as ELMo and BERT in the place of Word2Vec used in our model.

References

[1] P. Agarwal, J. Strötgen, L. Del Corro, J. Hoffart, and G. Weikum, "Dianed: Time-aware named entity disambiguation for diachronic corpora," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2018, pp. 686–693.

[2] E. Amitay, N. Har'El, R. Sivan, and A. Soffer, "Web-a-where: Geotagging web content," in *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, 2004, pp. 273–280.

[3] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, "Dbpedia: A nucleus for a web of open data," in *The semantic web*, Springer, 2007, pp. 722–735.

[4] A. Bagga and B. Baldwin, "Entity-based cross-document coreferencing using the vector space model," in *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, 1998, pp. 79–85.

[5] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[6] M. Ballesteros, C. Dyer, and N. A. Smith, "Improved transition-based parsing by modeling characters instead of words with lstms," *arXiv preprint arXiv:1508.00657*, 2015.

[7] D. S. Batista, B. Martins, and M. J. Silva, "Semi-supervised bootstrapping of relationship extractors with distributional semantics," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 499–504.

[8] M. Bevilacqua, T. Pasini, A. Raganato, R. Navigli, *et al.*, "Recent trends in word sense disambiguation: A survey," in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, International Joint Conference on Artificial Intelligence, Inc, 2021.

[9] R. Blanco, G. Ottaviano, and E. Meij, "Fast and space-efficient entity linking for queries," in *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, 2015, pp. 179–188.

[10] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: A collaboratively created graph database for structuring human knowledge," in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, 2008, pp. 1247–1250.

[11] S. Brin, "Extracting patterns and relations from the world wide web," in *International workshop on the world wide web and databases*, Springer, 1998, pp. 172–183.

[12] S. Broscheit, "Investigating entity knowledge in bert with simple neural end-to-end entity linking," *arXiv preprint arXiv:2003.05473*, 2020.

[13] R. Bunescu and M. Pasca, "Using encyclopedic knowledge for named entity disambiguation," 2006.

[14] A. E. Cano, G. Rizzo, A. Varga, M. Rowe, M. Stankovic, and A.-S. Dadzie, "Making sense of microposts:(# microposts2014) named entity extraction & linking challenge," in *Ceur workshop proceedings*, vol. 1141, 2014, pp. 54–60.

[15] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. Hruschka, and T. M. Mitchell, "Toward an architecture for never-ending language learning," in *Twenty-Fourth AAAI conference on artificial intelligence*, 2010.

[16] J. Chen, D. Ji, C. L. Tan, and Z.-Y. Niu, "Relation extraction using label propagation based semi-supervised learning," in *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, 2006, pp. 129–136.

[17] X. Cheng and D. Roth, "Relational inference for wikification," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 2013, pp. 1787–1796.

[18] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *Journal of machine learning research*, vol. 12, no. ARTICLE, pp. 2493–2537, 2011.

[19] S. Cucerzan, "Large-scale named entity disambiguation based on wikipedia data," in *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, 2007, pp. 708–716.

[20] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[21] M. Dredze, P. McNamee, D. Rao, A. Gerber, and T. Finin, "Entity disambiguation for knowledge base population," in *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, 2010, pp. 277–285.

[22] G. Durrett and D. Klein, "Easy victories and uphill battles in coreference resolution," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 2013, pp. 1971–1982.

[23] H. Elsahar, E. Demidova, S. Gottschalk, C. Gravier, and F. Laforest, "Unsupervised open relation extraction," in *European Semantic Web Conference*, Springer, 2017, pp. 12–16.

[24] Y. Eshel, N. Cohen, K. Radinsky, S. Markovitch, I. Yamada, and O. Levy, "Named entity disambiguation for noisy text," *arXiv preprint arXiv:1706.09147*, 2017.

[25] W. Fang, J. Zhang, D. Wang, Z. Chen, and M. Li, "Entity disambiguation by knowledge and text jointly embedding," in *Proceedings of the 20th SIGNLL conference on computational natural language learning*, 2016, pp. 260–269.

[26] Y. Fang and M.-W. Chang, "Entity linking on microblogs with spatial and temporal signals," *Transactions of the Association for Computational Linguistics*, vol. 2, pp. 259–272, 2014.

[27] Z. Fang, Y. Cao, Q. Li, D. Zhang, Z. Zhang, and Y. Liu, "Joint entity linking with deep reinforcement learning," in *The World Wide Web Conference*, 2019, pp. 438–447.

[28] S. Farazi and D. Rafiei, "Top-k frequent term queries on streaming data," in *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, IEEE, 2019, pp. 1582–1585.

[29] C. Fellbaum, "Wordnet," in *Theory and applications of ontology: computer applications*, Springer, 2010, pp. 231–243.

[30] I. P. Fellegi and A. B. Sunter, "A theory for record linkage," *Journal of the American Statistical Association*, vol. 64, no. 328, pp. 1183–1210, 1969.

[31] Z. Feng, Q. Wang, W. Jiang, Y. Lyu, and Y. Zhu, "Knowledge-enhanced named entity disambiguation for short text," in *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, 2020, pp. 735–744.

[32] P. Ferragina and U. Scaiella, "Fast and accurate annotation of short texts with wikipedia pages," *IEEE software*, vol. 29, no. 1, pp. 70–75, 2011.

[33] P. Ferragina and U. Scaiella, "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*, 2010, pp. 1625–1628.

[34] T. Févry, N. FitzGerald, L. B. Soares, and T. Kwiatkowski, "Empirical evaluation of pretraining strategies for supervised entity linking," *arXiv preprint arXiv:2005.14253*, 2020.

[35] J. R. Finkel, T. Grenager, and C. D. Manning, "Incorporating non-local information into information extraction systems by gibbs sampling," in *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, 2005, pp. 363–370.

[36] M. Fleischman and E. Hovy, "Fine grained classification of named entities," in *COLING 2002: The 19th International Conference on Computational Linguistics*, 2002.

[37] M. Francis-Landau, G. Durrett, and D. Klein, "Capturing semantic similarity for entity linking with convolutional neural networks," *arXiv preprint arXiv:1604.00734*, 2016.

[38] O.-E. Ganea and T. Hofmann, "Deep joint entity disambiguation with local neural attention," *arXiv preprint arXiv:1704.04920*, 2017.

[39] A. Globerson, N. Lazic, S. Chakrabarti, A. Subramanya, M. Ringaard, and F. Pereira, "Collective entity resolution with multi-focal attention," 2016.

[40] R. Grishman and B. M. Sundheim, "Message understanding conference-6: A brief history," in *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*, 1996.

[41] S. Guo, M.-W. Chang, and E. Kiciman, "To link or not to link? a study on end-to-end tweet entity linking," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2013, pp. 1020–1030.

[42] Z. Guo and D. Barbosa, "Robust named entity disambiguation with random walks," *Semantic Web*, vol. 9, no. 4, pp. 459–479, 2018.

[43] Z. Guo and D. Barbosa, "Robust entity linking via random walks," in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, 2014, pp. 499–508.

[44] N. Gupta, S. Singh, and D. Roth, "Entity linking via joint encoding of types, descriptions, and context," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 2681–2690.

[45] M. Habib and M. van Keulen, "Need4tweet: A twitterbot for tweets named entity extraction and disambiguation," in *Proceedings of ACL-IJCNLP 2015 System Demonstrations*, 2015, pp. 31–36.

[46] M. B. Habib and M. Van Keulen, "Unsupervised improvement of named entity extraction in short informal context using disambiguation clues.," in *SWAIE*, 2012, pp. 1–10.

[47] X. Han, L. Sun, and J. Zhao, "Collective entity linking in web text: A graph-based method," in *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, 2011, pp. 765–774.

[48] Z. He, S. Liu, M. Li, M. Zhou, L. Zhang, and H. Wang, "Learning entity representation for entity disambiguation," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2013, pp. 30–34.

[49] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[50] J. Hoffart, F. M. Suchanek, K. Berberich, and G. Weikum, "Yago2: A spatially and temporally enhanced knowledge base from wikipedia," *Artificial Intelligence*, vol. 194, pp. 28–61, 2013.

[51] J. Hoffart, M. A. Yosef, I. Bordino, *et al.*, "Robust disambiguation of named entities in text," in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, 2011, pp. 782–792.

[52] B. Huang, H. Wang, T. Wang, Y. Liu, and Y. Liu, "Entity linking for short text using structured knowledge graph via multi-grained text matching.," in *INTERSPEECH*, 2020, pp. 4178–4182.

[53] H. Huang, L. Heck, and H. Ji, "Leveraging deep neural networks and knowledge graphs for entity disambiguation," *arXiv preprint arXiv:1504.07678*, 2015.

[54] Z. Huang, W. Xu, and K. Yu, "Bidirectional lstm-crf models for sequence tagging," *arXiv preprint arXiv:1508.01991*, 2015.

[55] H. Ji, R. Grishman, H. Dang, K. Griffit, and J. Ellis, "Overview of the tac 2010 knowledge base population track," in *Proceedings of the 2010 Text Analysis Conference*, 2010.

[56] Y. Jiangwei and D. Rafiei, "Geotagging named entities in news and online documents," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, 2016, pp. 1321–1330.

[57] E. Kamalloo and D. Rafiei, "A coherent unsupervised model for toponym resolution," in *Proceedings of the 2018 World Wide Web Conference*, 2018, pp. 1287–1296.

[58] Y. Kim, Y. Jernite, D. Sontag, and A. M. Rush, "Character-aware neural language models," in *Thirtieth AAAI conference on artificial intelligence*, 2016.

[59] N. Kolitsas, O.-E. Ganea, and T. Hofmann, "End-to-end neural entity linking," *arXiv preprint arXiv:1808.07699*, 2018.

[60] G. Kondrak, "N-gram similarity and distance," in *International symposium on string processing and information retrieval*, Springer, 2005, pp. 115–126.

[61] S. Kulkarni, A. Singh, G. Ramakrishnan, and S. Chakrabarti, "Collective annotation of wikipedia entities in web text," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009, pp. 457–466.

[62] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," *arXiv preprint arXiv:1603.01360*, 2016.

[63] K. Lee, L. He, M. Lewis, and L. Zettlemoyer, "End-to-end neural coreference resolution," *arXiv preprint arXiv:1707.07045*, 2017.

[64] D. B. Lenat, "Cyc: A large-scale investment in knowledge infrastructure," *Communications of the ACM*, vol. 38, no. 11, pp. 33–38, 1995.

[65] T. Lin, O. Etzioni, *et al.*, "No noun phrase left behind: Detecting and typing unlinkable entities," in *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, 2012, pp. 893–903.

[66] Y. Lin, S. Shen, Z. Liu, H. Luan, and M. Sun, "Neural relation extraction with selective attention over instances," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016, pp. 2124–2133.

[67] W. Ling, T. Luis, L. Marujo, *et al.*, "Finding function in form: Compositional character models for open vocabulary word representation," *arXiv preprint arXiv:1508.02096*, 2015.

[68] B. Locke and J. Martin, "Named entity recognition: Adapting to microblogging," *Senior Thesis, University of Colorado*, 2009.

[69] L. Logeswaran, M.-W. Chang, K. Lee, K. Toutanova, J. Devlin, and H. Lee, "Zero-shot entity linking by reading entity descriptions," *arXiv preprint arXiv:1906.07348*, 2019.

[70] D. Marcheggiani and I. Titov, "Discrete-state variational autoencoders for joint discovery and factorization of relations," *Transactions of the Association for Computational Linguistics*, vol. 4, pp. 231–244, 2016.

[71] E. Meij, W. Weerkamp, and M. De Rijke, "Adding semantics to microblog posts," in *Proceedings of the fifth ACM international conference on Web search and data mining*, 2012, pp. 563–572.

[72] P. N. Mendes, M. Jakob, A. Garcıa-Silva, and C. Bizer, "Dbpedia spotlight: Shedding light on the web of documents," in *Proceedings of the 7th international conference on semantic systems*, 2011, pp. 1–8.

[73] R. Mihalcea and A. Csomai, "Wikify! linking documents to encyclopedic knowledge," in *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, 2007, pp. 233–242.

[74] T. Mikolov, W.-t. Yih, and G. Zweig, "Linguistic regularities in continuous space word representations," in *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*, 2013, pp. 746–751.

[75] D. Milne and I. H. Witten, "Learning to link with wikipedia," in *Proceedings of the 17th ACM conference on Information and knowledge management*, 2008, pp. 509–518.

[76] M. Mintz, S. Bills, R. Snow, and D. Jurafsky, "Distant supervision for relation extraction without labeled data," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, 2009, pp. 1003–1011.

[77] S. Mishra and J. Diesner, "Semi-supervised named entity recognition in noisy-text," in *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, 2016, pp. 203–212.

[78] M. Miwa and M. Bansal, "End-to-end relation extraction using lstms on sequences and tree structures," *arXiv preprint arXiv:1601.00770*, 2016.

[79] A. Moro, A. Raganato, and R. Navigli, "Entity linking meets word sense disambiguation: A unified approach," *Transactions of the Association for Computational Linguistics*, vol. 2, pp. 231–244, 2014.

[80] N. Nakashole, T. Tylenda, and G. Weikum, "Fine-grained semantic typing of emerging entities," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2013, pp. 1488–1497.

[81] D. B. Nguyen, M. Theobald, and G. Weikum, "J-reed: Joint relation extraction and entity disambiguation," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017, pp. 2227–2230.

[82] T. H. Nguyen, N. R. Fauceglia, M. R. Muro, O. Hassanzadeh, A. Gliozzo, and M. Sadoghi, "Joint learning of local and global features for entity linking via neural networks," in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 2016, pp. 2310–2320.

[83] T. H. Nguyen and R. Grishman, "Relation extraction: Perspective from convolutional neural networks," in *Proceedings of the 1st workshop on vector space modeling for natural language processing*, 2015, pp. 39–48.

[84] Y. Onoe and G. Durrett, "Fine-grained entity typing for domain independent entity linking," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 2020, pp. 8576–8583.

[85] S. M. Paradesi, "Geotagging tweets using their content," in *Twenty-Fourth International FLAIRS Conference*, 2011.

[86] H. Paulheim, "Knowledge graph refinement: A survey of approaches and evaluation methods," *Semantic web*, vol. 8, no. 3, pp. 489–508, 2017.

[87] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[88] M. Pershina, Y. He, and R. Grishman, "Personalized page rank for named entity disambiguation," in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2015, pp. 238–243.

[89] M. E. Peters, W. Ammar, C. Bhagavatula, and R. Power, "Semi-supervised sequence tagging with bidirectional language models," *arXiv preprint arXiv:1705.00108*, 2017.

[90] M. E. Peters, M. Neumann, R. L. Logan IV, *et al.*, "Knowledge enhanced contextual word representations," *arXiv preprint arXiv:1909.04164*, 2019.

[91] M. E. Peters, M. Neumann, M. Iyyer, *et al.*, "Deep contextualized word representations," in *Proc. of NAACL*, 2018.

[92] M. C. Phan, A. Sun, Y. Tay, J. Han, and C. Li, "Neupl: Attention-based semantic matching and pair-linking for entity disambiguation," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017, pp. 1667–1676.

[93] A. Rahman and V. Ng, "Inducing fine-grained semantic classes via hierarchical and collective classification," in *Proceedings of the 23rd international conference on computational linguistics (Coling 2010)*, 2010, pp. 931–939.

[94] J. Raiman and O. Raiman, "Deeptype: Multilingual entity linking by neural type system evolution," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018.

[95] L. Ratinov, D. Roth, D. Downey, and M. Anderson, "Local and global algorithms for disambiguation to wikipedia," in *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, 2011, pp. 1375–1384.

[96] M. P. K. Ravi, K. Singh, I. O. Mulang, S. Shekarpour, J. Hoffart, and J. Lehmann, "Cholan: A modular approach for neural entity linking on wikipedia and wikidata," *arXiv preprint arXiv:2101.09969*, 2021.

[97] A. Sakor, I. O. Mulang, K. Singh, *et al.*, "Old is gold: Linguistic driven approach for entity and relation linking of short text," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 2336–2346.

[98] E. Sandhaus, "New york times corpus: Corpus overview," *LDC catalogue entry LDC2008T19*, 2008.

[99] E. F. Sang and F. De Meulder, "Introduction to the conll-2003 shared task: Language-independent named entity recognition," *arXiv preprint cs/0306050*, 2003.

[100] O. Sevgili, A. Shelmanov, M. Arkhipov, A. Panchenko, and C. Biemann, "Neural entity linking: A survey of models based on deep learning," *arXiv preprint arXiv:2006.00575*, 2020.

[101] M. Shang, T. Wang, M. Eric, *et al.*, "Entity resolution in open-domain conversations," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Papers*, 2021, pp. 26–33.

[102] W. Shen, J. Wang, and J. Han, "Entity linking with a knowledge base: Issues, techniques, and solutions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 2, pp. 443–460, 2014.

[103] A. Sil, G. Kundu, R. Florian, and W. Hamza, "Neural cross-lingual entity linking," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[104] E. Simon, V. Guigue, and B. Piwowarski, "Unsupervised information extraction: Regularizing discriminative approaches with relation distribution losses," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 1378–1387.

[105] D. Sorokin and I. Gurevych, "Mixing context granularities for improved entity linking on question answering data across entity categories," *arXiv preprint arXiv:1804.08460*, 2018.

[106] V. I. Spitkovsky and A. X. Chang, "A cross-lingual dictionary for english wikipedia concepts," 2012.

[107] M. Srinivasan and D. Rafiei, "Location-aware named entity disambiguation," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 3433–3438.

[108] Y. Sun, L. Lin, D. Tang, N. Yang, Z. Ji, and X. Wang, "Modeling mention, context and entity with neural networks for entity disambiguation," in *Twenty-fourth international joint conference on artificial intelligence*, 2015.

[109] C. Tan, F. Wei, P. Ren, W. Lv, and M. Zhou, "Entity linking for queries by searching wikipedia sentences," *arXiv preprint arXiv:1704.02788*, 2017.

[110] R. Usbeck, A.-C. N. Ngomo, M. Röder, *et al.*, "Agdistis-graph-based disambiguation of named entities using linked data," in *International semantic web conference*, Springer, 2014, pp. 457–471.

[111] R. Usbeck, M. Röder, A.-C. Ngonga Ngomo, *et al.*, "Gerbil: General entity annotator benchmarking framework," in *Proceedings of the 24th international conference on World Wide Web*, 2015, pp. 1133–1143.

[112] J. Waitelonis and H. Sack, "Named entity linking in# tweets with kea.," in *# Microposts*, 2016, pp. 61–63.

[113] X. Wang, Y. Jiang, N. Bach, *et al.*, "Automated concatenation of embeddings for structured prediction," *arXiv preprint arXiv:2010.05006*, 2020.

[114] L. Wu, F. Petroni, M. Josifoski, S. Riedel, and L. Zettlemoyer, "Scalable zero-shot entity linking with dense entity retrieval," *arXiv preprint arXiv:1911.03814*, 2019.

[115] I. Yamada, H. Shindo, H. Takeda, and Y. Takefuji, "Joint learning of the embedding of words and entities for named entity disambiguation," *arXiv preprint arXiv:1601.01343*, 2016.

[116] Y. Yang and M.-W. Chang, "S-mart: Novel tree-based structured learning algorithms applied to tweet entity linking," *arXiv preprint arXiv:1609.08075*, 2016.

[117] H.-T. Zhang, M.-L. Huang, and X.-Y. Zhu, "A unified active learning framework for biomedical relation extraction," *Journal of Computer Science and Technology*, vol. 27, no. 6, pp. 1302–1313, 2012.

[118] H. Zhang, X. Zhao, and Y. Song, "A brief survey and comparative study of recent development of pronoun coreference resolution," *arXiv preprint arXiv:2009.12721*, 2020.

[119] L. Zhang and A. Rettinger, "X-lisa: Cross-lingual semantic annotation," *Proceedings of the VLDB Endowment*, vol. 7, no. 13, pp. 1693–1696, 2014.

[120] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," *Advances in neural information processing systems*, vol. 28, pp. 649–657, 2015.

[121] Y. Zhang, P. Qi, and C. D. Manning, "Graph convolution over pruned dependency trees improves relation extraction," *arXiv preprint arXiv:1809.10185*, 2018.

[122] Y. Zhang, V. Zhong, D. Chen, G. Angeli, and C. D. Manning, "Position-aware attention and supervised data improve slot filling," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 35–45.

[123] A. Zhila, W.-t. Yih, C. Meek, G. Zweig, and T. Mikolov, "Combining heterogeneous models for measuring relational similarity," in *Proceedings of the 2013 conference of the North American chapter of the association for computational linguistics: Human language technologies*, 2013, pp. 1000–1009.

[124] Y. Zhou, L. Nie, O. Rouhani-Kalleh, F. Vasile, and S. Gaffney, "Resolving surface forms to wikipedia topics," in *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, 2010, pp. 1335–1343.

[125] S. Zwicklbauer, C. Seifert, and M. Granitzer, "Robust and collective entity disambiguation through semantic embeddings," in *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, 2016, pp. 425–434.