上海交通大學

SHANGHAI JIAO TONG UNIVERSITY

学士学位论文

THESIS OF BACHELOR

论文题目：**Commonsense Knowledge Graph Representation and Construction**

学生姓名：林禹臣

学生学号：5140309507

专　　业：计算机科学与技术 (IEEE)

指导教师：朱其立 教授

学院 (系)：电子信息与电气工程学院

Commonsense Knowledge Graph Representation and Construction

摘　要

在人工智能领域，常识性知识指的是一个由人类世界日常生活中的事实组成的集合，比如 ‘‘冰是凉的” (属性关系)，‘‘咀嚼是吃饭的子动作之一”(子动作关系)，‘‘桌子和椅子常常摆在一起”(近邻关系)，等等。常识性知识有关的研究一直是人工智能领域的热点之一，它可以向人工智能系统和各类模型提供物理世界中丰富的背景知识，从而提升性能和效率。这类先验知识可以为很多下游任务提供支持和改善的空间，比如自然语言处理 (NLP) 中的文本推理 (textual inference) 任务，和计算机视觉 (CV) 中的物体检查 (object detection) 任务。主流研究中主要采用三元组 (triple) 的形式对常识性知识进行表示，典型的例子就是 ConceptNet，一个起源于麻省理工大学的当今最大的常识性知识库。

绝大部分的常识性知识库都是通过人类手工标注构造完成的，来自众包社区的力量是因素，这导致常识性知识库通常很难迅速扩大。除此之外，标注者数量通常有限所以多样性不足，人力成本也十分昂贵，这些因素导致了传统构建常识性知识库存在的问题和缺陷。因此，一个自动化构建常识性知识库的方案会非常有益于这个领域乃至人工智能领域的发展。近邻关系 (LocatedNear) 是一种描述两个物体在现实物理世界的空间中常常出现在一起的常识性关系，在 ConceptNet 中却只有 49 个三元组。笔者在这篇毕业论文的第一部分以这一常识关系为例，研究了如何从海量文本中抽取常识性关系，从而实现扩增常识性数据库。笔者从句子级别的关系分类器入手，研究了如何通过提升分类器的聚合分类器预测结果来提高常识关系抽取的效果。此外，本文还提出了两个用于测试常识关系抽取的标准数据集：第一个是具有关系分类标准的 5000 句子，每个句子被标注是否这句话是否描述了常识性关系；第二个数据集则是语料库级别的 500 个三元组，用来检测关系抽取模型的效果。本文提出了若干方法来解决常识关系抽取任务，并且将这些方法与通用的关系抽取模型进行了比较。

假使我们已经拥有了一个相对比较完整的常识知识库，如何去表征它们依然是一个非常有挑战性的任务。近些年来，知识图谱的嵌入式表达 (KGE) 非常流行，有很多模型被提出，典型的有基于转移的 TransE 模型和基于语意匹配的 ANALOGY 模型。不过，这些模型大多都是为了普通的事实型知识图谱提出的，比如 FreeBase 和 DBPedia，或者为了词典型数据库 WordNet。而常识知识图谱有着自己本身的特点和困难，这些是普通的模型并没有关注过的。目前几乎没有专为常识知识图谱 (CSKG) 提出的表征模型。本论文第二部分主要围绕如何构造第一个 CSKGE 标准数据集而展开，并通过实验比较了当前的主流 KGE 模型在这个数据集上的效果。作者还针对 CSKG 的特点提出了一种新颖的 CSKGE 模型对常识知识图谱进行了表征，取得了超越主流模型的效果。

另外，不同文化之间发生的概念迁移也是一种非常重要的常识。针对概念的跨文化异同对扩展常识性知识库是非常有帮助的一种信息，这种文化变差也是跨语言的自然语言理解中重要的一环，尤其是当我们在处理社交媒体数据时。比如，不同文化的人们对同一命名实体可能会有着截然不同的看法和观点；再如，理解另一个语言中的俚语、网络流行语需要能够理解跨语言、跨文化的语言相似概念。这篇论文的第三部分主要围绕如何从社交网络中挖掘跨文化异同展开，提出了一个轻量

级但高效率的模型，并且利用两个新颖的任务去测评：1) 挖掘社交媒体中的对于命名实体跨文化差异；2) 翻译网络流行语、俚语。通过实验分析，本文提出的模型效果超过其他基准方法。它可以为跨文化、跨语言的计算社会学研究提供基础的计算模型，也会使机器翻译模型受益。

关键词：常识知识图谱　知识图谱　知识图谱嵌入　关系抽取　文化差异　社交媒体　机器翻译

COMMONSENSE KNOWLEDGE GRAPH REPRESENTATION AND CONSTRUCTION

ABSTRACT

Commonsense knowledge can be defined as a set of facts about our everyday world, such as *ice is cold* (HASPROPERTY), *chewing is a sub-event of eating* (HASSUBEVENT), *chair and table are typically found near each other* (LOCATEDNEAR), etc. Commonsense knowledge and related works have been one of the most important areas in Artificial Intelligence, because a lot of artificial intelligent systems can benefit from incorporating commonsense knowledge as background priors in their models. These kinds of commonsense facts have been used in many downstream tasks, such as textual entailment in Natural Language Processing (NLP) and object detection in Computer Vision (CV). The commonsense knowledge is often represented as relation triples in Common-Sense Knowledge Graphs (CSKGs), such as *ConceptNet*, one of the largest commonsense knowledge graphs available today.

Most commonsense knowledge bases are manually curated or crowd-sourced by community efforts and thus do not scale well. Another problem is that such commonsense knowledge bases are typically contributed by just a very limited number of people due to the cost of manual labor. Therefore, an automatic method of extracting commonsense relationship from textual corpora or other data is an essential topic. LOCATEDNEAR relation is a kind of commonsense knowledge describing two physical objects that are typically found near each other in real life, of which ConceptNet contains only 49 triples. In the first section of this thesis, the author studies how to automatically extract such relationship through a sentence-level relation classifier and aggregating the scores of entity pairs from a large corpus. Apart from that, we release two benchmark datasets for evaluation and future research: 1）one containing 5,000 sentences annotated with whether a mentioned entity pair has LOCATEDNEAR relation in the given sentence or not; 2）the other containing 500 pairs of physical objects and whether they are commonly located nearby. We propose a number of baseline methods for the tasks and compare the results with a state-of-the-art general-purpose relation classifier.

Even when we have a relatively complete commonsense knowledge graph, the representation of CSKGs is also a challenging task. Recently, Knowledge Graph Embedding (KGE) techniques are so trendy and thus many models were proposed, from simple translational model like TransE to semantic-matching model like ANALOGY. Nevertheless, these models are mostly designed for factual knowledge bases and lexical databases such as FreeBase (FB15K) and WordNet (WN18) as well as DBPedia. CSKGs have a lot of specific features that are significant different from above-mentioned graphs. To the best of our knowledge, there is no existing dataset and model to investigate KGE for CSKGs, which we denote as CSKGE (Common-Sense Knowledge Graph Embedding). The second section of this thesis proposes the very first dataset for CSKGE

and investigate the characteristics as well as the performance of state-of-the-art KGE models on it. The author also proposes a novel CSKGE model purposely designed for CSKGs.

Additionally, the phenomenon of concept drifting across different cultures is also a important type of commonsense knowledge. Cross-cultural differences and similarities of concepts are helpful in extending multilingual CSKGs. The culture shift in concept understanding is common in cross-lingual natural language understanding, especially for research in social media. For instance, people of distinct cultures often hold different opinions on a single named entity. Also, understanding slang terms across languages requires knowledge of cross-cultural similarities. The third section of this thesis studies the problem of computing such cross-cultural differences and similarities. This thesis presents a lightweight yet effective approach, and evaluate it on two novel tasks: 1) mining cross-cultural differences of named entities and 2) finding similar terms for slang across languages. Experimental results show that our framework substantially outperforms a number of baseline methods on both tasks. The framework could be useful for machine translation applications and research in computational social science.

KEY WORDS: commonsense knowledge, knowledge graph, commonsense knowledge graph, knowledge graph embedding, relation extraction, cultural differences, social media, machine translation

Contents

List of Figures

List of Tables

Chapter 1 Introduction

Commonsense knowledge can be defined as a set of facts about our everyday world, such as *ice is cold* (HASPROPERTY), *chewing is a sub-event of eating* (HASSUBEVENT), *chair and table are typically found near each other* (LOCATEDNEAR), etc. Commonsense knowledge and related works have been one of the most important areas in Artificial Intelligence, because a lot of artificial intelligent systems can benefit from incorporating commonsense knowledge as background priors in their models. These kinds of commonsense facts have been used in many downstream tasks, such as textual entailment in Natural Language Processing (NLP) and object detection in Computer Vision (CV). The commonsense knowledge is often represented as relation triples in Common-Sense Knowledge Graphs (CSKGs), such as *ConceptNet*, one of the largest commonsense knowledge graphs available today.

1.1 Common-Sense Relation Extraction

Most commonsense knowledge bases are manually curated or crowd-sourced by community efforts and thus do not scale well. Another problem is that such commonsense knowledge bases are typically contributed by just a very limited number of people due to the cost of manual labor. Therefore, an automatic method of extracting commonsense relationship from textual corpora or other data is an essential topic. LOCATEDNEAR relation is a kind of commonsense knowledge describing two physical objects that are typically found near each other in real life, of which ConceptNet contains only 49 triples. In the first section of this thesis, the author studies how to automatically extract such relationship through a sentence-level relation classifier and aggregating the scores of entity pairs from a large corpus. Apart from that, we release two benchmark datasets for evaluation and future research: 1）one containing 5,000 sentences annotated with whether a mentioned entity pair has LOCATEDNEAR relation in the given sentence or not; 2）the other containing 500 pairs of physical objects and whether they are commonly located nearby. We propose a number of baseline methods for the tasks and compare the results with a state-of-the-art general-purpose relation classifier.

Commonsense knowledge is an important ingredient in machine comprehension and inference. Artificial intelligence systems can benefit from incorporating commonsense knowledge as background, such as *ice is cold* (HASPROPERTY), *chewing is a sub-event of eating* (HASSUBEVENT), *chair and table are typically found near each other* (LOCATEDNEAR), etc. These kinds of commonsense facts have been used in many downstream tasks, such as textual entailment [1, 2] and visual recognition tasks [3].

The commonsense knowledge is often represented as relation triples in commonsense knowledge bases, such as *ConceptNet* [4], one of the largest commonsense knowledge graphs available today. However, most commonsense knowledge bases are manually curated or crowd-sourced by community efforts and thus do not scale well. For example, ConceptNet contains only 49 LOCATEDNEAR relation triples. Another problem is that such commonsense knowledge bases are typically contributed by just a very limited number of people due to the cost of manual labor. Thus no meaningful statistical scores can be associated with the triples, making

Figure 1–1 LocatedNear facts assist the detection of vague objects: if a set of knife, fork and plate is on the table, one may believe there is a glass beside based on the commonsense, even though these objects are hardly visible due to low light.

rank-based computation difficult. For instance, although ConceptNet gives a confidence score (from 0 to infinity) to each triple, most of the triples have the default score of 1, simply because the human contributor did not or could not provide a score. If such commonsense knowledge is harnessed automatically from open-domain text corpora, both of the above problems can be effectively addressed. Open information extraction not only provides the much needed scale, but also valuable statistics that can turn into confidence scores.

This paper aims to automatically extract the commonsense LocatedNear relation between physical objects from textual corpora. LocatedNear is defined as the relationship between two objects typically found near each other in real life. [1] We focus on LocatedNear relation for these reasons:

- LocatedNear facts provide helpful prior knowledge to object detection tasks in complex image scenes [5]. See Figure 1–1 for an example.
- This commonsense knowledge can benefit reasoning related to spatial facts and physical scenes in reading comprehension, question answering, etc. [6]
- Existing knowledge bases have very few facts for this relation (*ConceptNet 5.5* has only 49 triples of LocatedNear relation).

(The guess can be based on commonsense knowledge learned from room settings scene description in articles and texts.) Such prior knowledge helps with the object detection accuracy;

1. LocatedNearrelation is useful for object detection in complex image scenes. For example, in a dimly lit room with a dining table and some chairs. One may guess that plates and other kitchenware

[1] Because some physical objects can be a location itself, this relation may include some instances of the atLocation relation, e.g., *room* and *door*.

maybe present on the table. Such prior knowledge helps with the object detection accuracy.

2. LocatedNearrelation can also be useful for automated conversation systems where meaningful context maybe added to the conversation.
3. LocatedNearrelation can benefit general reasoning in reading comprehension, question answering and many other AI tasks.
4. Existing knowledge bases such as Concept Net has very limited facts for this relation.

Automatic extraction of relations from open text has a short but rich history. Attempts have been made to extract isA, causal, correlation, and also open domain relations (e.g., ReVerb, Yago). LocatedNear relation is unique and poses significant challenges for the following reasons: i) It involves physical (often visible) objects whereas other popular relations involve general concepts or just natural language terms. ii) The distribution of LocatedNear relation is not even across domains: it is more prevalent in literary work such as stories and dramas which come with descriptive scenes rather than in news, science & technology related articles or online user generated content. iii) Sentences in literature are often complex and nuanced, which makes extraction particularly challenging. Consider the objects "bed" and "star" in the following sentence: "*Until at last all the promenaders had gone home to bed, and I was alone with the star.*" Bed is not near the star because it's at another location. iv) Labeling such sentence is a non-trivial task, and obtaining a large training set is difficult and expensive.

Since raw text of novels tend to contain many descriptions of scene in real life, we argue that it is feasible to obtain unseen LocatedNear relations from raw novel text. We propose two novel tasks in extracting LocatedNear relation from textual corpora. One is a sentence-level relation classification problem which judges whether or not a sentence describes two objects (mentioned in the sentence) being physically close by. The other task is to produce a ranked list of LocatedNear facts with the given classified results of large number of sentences. We believe both two tasks can be used to automatically populate and complete existing commonsense knowledge bases. Notice that two objects that are *co-located* in a couple of sentences may not mean they have the LocatedNear relation as commonsense. Additionally, we create two benchmark datasets for evaluating LocatedNear relation extraction systems on the two tasks: one is 5,000 sentences each describing a scene of two physical objects and with a label indicating if the two objects are co-located in the scene; the other consists of 500 pairs of objects with human-annotated scores indicating confidences that a certain pair of objects are commonly located near in real life.[1]

1.2 Common-Sense Knowledge Graph Embedding

Even when we have a relatively complete commonsense knowledge graph, the representation of CSKGs is also a challenging task. Recently, Knowledge Graph Embedding (KGE) techniques are so trendy and thus many models were proposed, from simple translational model like TransE to semantic-matching model like ANALOGY. Nevertheless, these models are mostly designed for factual knowledge bases and lexical databases such as FreeBase (FB15K) and WordNet (WN18) as well as DBPedia. CSKGs have a lot of specific features that are significant different from above-mentioned graphs. To the best of our knowledge, there is

[1] https://github.com/adapt-sjtu/commonsense-locatednear

no existing dataset and model to investigate KGE for CSKGs, which we denote as CSKGE (Common-Sense Knowledge Graph Embedding). The second section of this thesis proposes the very first dataset for CSKGE and investigate the characteristics as well as the performance of state-of-the-art KGE models on it. The author also proposes a novel CSKGE model purposely designed for CSKGs.

Knowledge graphs such as Freebase [7] and WordNet [8] play an important role in building various artificial intelligence systems including question answering and personal assistant (Siri). These knowledge graphs store facts with a large amount of triplets in the form of (h, r, t), where r is the directed relation edge indicating the relationship between the left node h and the right node t. In this paper, we aim to learn low-dimensional vector representations for nodes (entities) and edges (relations) in a knowledge graph, which can be used to infer new facts.

Many successful methods have been proposed for knowledge graph representation learning in the past few years. Representative approaches include the translation-based models [9–11] and the bilinear models [12, 13]. Such approaches can well capture structural properties of knowledge graphs but largely ignore the textual information related with the nodes, which could be regarded as good supplementary feature for knowledge graphs especially those entities with few facts.

Recently, some methods [14–16] has been proposed to address this issue, utilizing the textual descriptions about nodes. However, the descriptions are different. For example, in FB15k, some nodes have long descriptions (343 words) and some nodes have very short descriptions (shorter than 3 words) or no descriptions[15]. Besides, some knowledge graphs may not have existing description about nodes to use. For example, ConceptNet [17] is a type of knowledge graphs that consists of commonsense knowledge only, which is also important in various artificial intelligence applications [18, 19] but not the main focus in previous knowledge graph representation learning. Unlike WordNet and Freebase, each node in ConceptNet is a textual phrase with an arbitrary number of words. And there is no existing description about such nodes.

On the other hand, the optimal combination of the structural and textual representations is not well studied in many previous methods [14, 15, 20], in which the structural representation and textual representation are aligned on separate loss function. A good representation of an entity should jointly encode both structure and text information. [16]

In this paper, we utilize the textual information conveyed by the nodes as the supplementary features, which allow the sharing of statistical strength between the words describing each node [21]. And we explore several simple and general approaches to combine textual features from the nodes as well as the structural information from the graph. Specifically, we consider two ways to capture the textual information that resides within the node. We also consider two approaches to integrate textual and structural information for learning node representations. Our experiments are conducted on three datasets. Two of them are public benchmark datasets that are the subsets of WordNet and Freebase. The third one is a dataset created from ConceptNet following similar procedure describe in [22]. Experimental results on the datasets show that our approach outperform baseline approaches.

Figure 1–2 Two social media messages about Nagoya from different cultures in 2012

1.3 Mining Common-Sense Concept Drift across Cultures

Additionally, the phenomenon of concept drifting across different cultures is also a important type of common-sense knowledge. Cross-cultural differences and similarities of concepts are helpful in extending multilingual CSKGs. The culture shift in concept understanding is common in cross-lingual natural language understanding, especially for research in social media. For instance, people of distinct cultures often hold different opinions on a single named entity. Also, understanding slang terms across languages requires knowledge of cross-cultural similarities. The third section of this thesis studies the problem of computing such cross-cultural differences and similarities. This thesis presents a lightweight yet effective approach, and evaluate it on two novel tasks: 1) mining cross-cultural differences of named entities and 2) finding similar terms for slang across languages. Experimental results show that our framework substantially outperforms a number of baseline methods on both tasks. The framework could be useful for machine translation applications and research in computational social science.

Computing similarities between terms is one of the most fundamental computational tasks in natural language understanding. Much work has been done in this area, most notably using the distributional properties drawn from large monolingual textual corpora to train vector representations of words or other linguistic units [23, 24]. However, computing cross-cultural similarities of terms between different cultures is still an open research question, which is important in cross-lingual natural language understanding. In this paper, we address cross-cultural research questions such as these:

1. *Were there any cross-cultural differences between Nagoya (a city in Japan) for native English speakers and 名古屋 (Nagoya in Chinese) for Chinese people in 2012?*
2. *What English terms can be used to explain "浮云" (a Chinese slang term)?*

These kinds of questions about cross-cultural differences and similarities are important in cross-cultural social studies, multi-lingual sentiment analysis, culturally sensitive machine translation, and many other NLP tasks, especially in social media. We propose two novel tasks in mining them from social media.

The first task is to mine cross-cultural differences in the perception of named entities (e.g., persons, places and organizations). Back in 2012, in the case of "Nagoya", many native English speakers posted their

pleasant travel experiences in Nagoya on Twitter. However, Chinese people overwhelmingly greeted the city with anger and condemnation on *Weibo* (a Chinese version of Twitter), because the city mayor denied the truthfulness of the Nanjing Massacre. Figure 1–2 illustrates two example microblog messages about Nagoya in Twitter and Weibo respectively.

The second task is to find similar terms for slang across cultures and languages. Social media is always a rich soil where slang terms emerge in many cultures. For example, ''浮云" literally means "floating clouds", but now almost equals to "nothingness" on the Chinese web. Our experiments show that well-known online machine translators such as *Google Translate* are only able to translate such slang terms to their literal meanings, even under clear contexts where slang meanings are much more appropriate.

Enabling intelligent agents to understand such cross-cultural knowledge can benefit their performances in various cross-lingual language processing tasks. Both tasks share the same core problem, which is **how to compute cross-cultural differences (or similarities) between two terms from different cultures.** A term here can be either an ordinary word, an entity name, or a slang term. We focus on names and slang in this paper for they convey more social and cultural connotations.

There are many works on cross-lingual word representation [25] to compute general cross-lingual similarities [26]. Most existing models require bilingual supervision such as aligned parallel corpora, bilingual lexicons, or comparable documents [27–29]. However, they do not purposely preserve social or cultural characteristics of named entities or slang terms, and the required parallel corpora are rare and expensive.

In this paper, we propose a lightweight yet effective approach to project two incompatible monolingual word vector spaces into a single bilingual word vector space, known as social vector space (*SocVec*). A key element of SocVec is the idea of "bilingual social lexicon", which contains bilingual mappings of selected words reflecting psychological processes, which we believe are central to capturing the socio-linguistic characteristics. Our contribution in this paper is three-fold:

1. We present an effective approach (SocVec) to mine cross-cultural similarities and differences of terms, which could benefit research in machine translation, cross-cultural social media analysis, and other cross-lingual research in natural language processing and computational social science.
2. We propose two novel and important tasks in cross-cultural social studies and social media analysis. Experimental results on our annotated datasets show that the proposed method outperforms many strong baseline methods.
3. We release a prototype tool for the proposed approach, two datasets on the above tasks, and several resources, which could potentially benefit future research in cross-cultural social studies and social media analysis.

Chapter 2 Automatic Extraction of Commonsense LocatedNear Knowledge

2.1 Sentence-level LocatedNear Relation Classification

Problem Statement Given a sentence s mentioning a pair of physical objects $<e_i, e_j>$, we call $<s, e_i, e_j>$ an *instance*. For each instance, the problem is to determine whether e_i and e_j are located near each other in the physical scene described in the sentence s. For example, suppose e_i is "dog", e_j is "cat", and s = "*The King puts his dog and cat on the table.*". As it is true that the two objects are located near in this sentence, a successful classification model is expected to label this instance as *True*. However, if s_2 = "*My dog is older than her cat.*", then the label of the instance $<s_2, e_i, e_j>$ is *False*, because s_2 just talks about a comparison in age. In the following subsections, we present two different kinds of baseline methods for this binary classification task: feature-based methods and LSTM-based neural architectures.

2.1.1 Feature-based Methods

Our first baseline method is an SVM classifier based on following features commonly used in many relation extraction models [30]:

- *Bag of Words (BW)*: the set of words that ever appeared in the sentence.
- *Bag of Path Words (BPW)*: the set of words that appeared on the shortest dependency path between objects e_i and e_j in the dependency tree of the sentence s, plus the words in the two subtrees rooted at e_i and e_j in the tree.
- *Bag of Adverbs and Prepositions (BAP)*: the existence of adverbs and prepositions in the sentence as binary features.
- *Global Features (GF)*: the length of the sentence, the number of nouns, verbs, adverbs, adjectives, determiners, prepositions and punctuations in the whole sentence.
- *Shortest Dependency Path features (SDP)*: the same features as with GF but in dependency parse trees of the sentence and the shortest path between e_i and e_j, respectively.
- *Semantic Similarity features (SS)*: the cosine similarities between the pre-trained *GloVe* word embeddings [23] of the two object words.

We evaluate *linear* and *RBF* kernels with different parameter settings, and find the *RBF* kernel with $\{C = 100, \gamma = 10^{-3}\}$ performs the best overall.

2.1.2 LSTM-based Neural Architectures

We observe that the existence of LocatedNear relation in an instance $<s,e_1,e_2>$ depends on two major information sources: one is from the semantic and syntactical features of sentence s and the other is from the object pair $<e_1,e_2>$. By this intuition, we design our LSTM-based model with two parts, shown in lower part

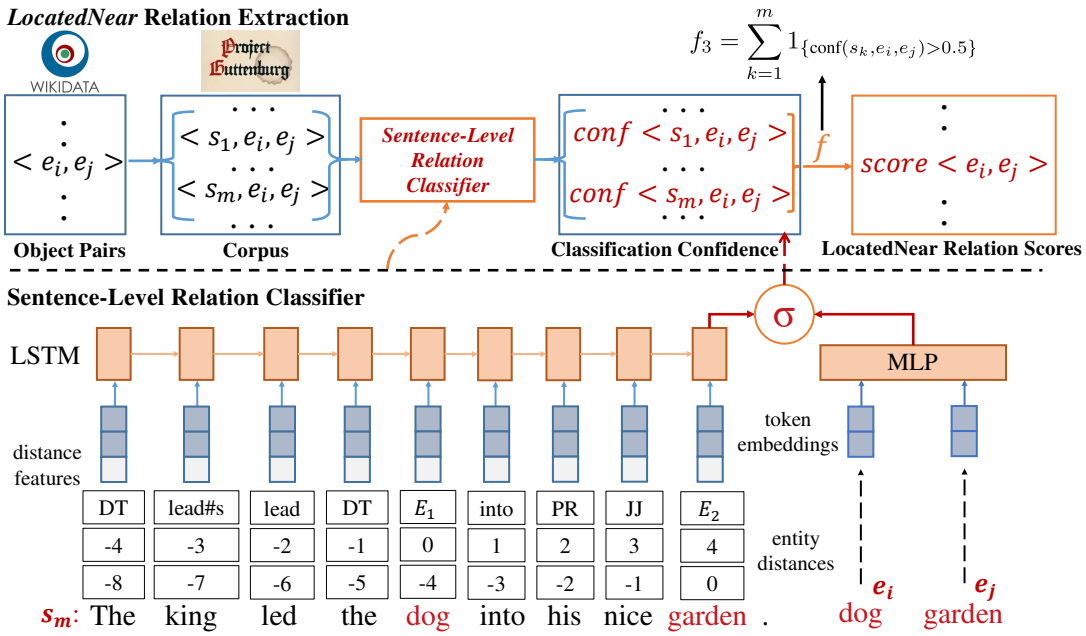

Figure 2–1 Framework with a LSTM-based classifier

Table 2–1 Examples of four types of tokens during sentence normalization. (#s stands for subjects and #o for objects)

Level	Examples
Objects	E_1, E_2
Lemma	open, lead, into, ...
Dependency Role	open#s, open#o, into#o, ...
POS Tag	DT, PR, CC, JJ, ...

of Figure 2–1. The left part is for encoding the syntactical and semantic information of the sentence s, while the right part is encoding the semantic similarity between the pre-trained word embeddings of e_1 and e_2.

Solely relying on the original word sequence of a sentence s has two problems: (i) the irrelevant words in the sentence can introduce noise into the model; (ii) the large vocabulary of original sentences induce too many parameters, which may cause over-fitting. For example, given two sentences "*The king led the dog into his nice garden.*" and "*A criminal led the dog into a poor garden.*". The object pair is <*dog, garden*> in both sentences. The two words "*lead*" and "*into*" are essential for determining whether the object pair is located near, but they are not attached with due importance. Also, the semantic differences between irrelevant words, such as "king" and "criminal", "beautiful" and "poor", are not useful to the co-location relation between the "dog" and "garden", and thus tend to act as noise.

To address the above issues, we propose a normalized sentence representation method merging the three most important and relevant kinds of information about each instance: lemmatized forms, POS (Part-of-Speech) tags and dependency roles. We first replace the two nouns in the object pair as "E_1" and "E_2", and

keep the lemmatized form of the original words for all the *verbs, adverbs and prepositions*, which are highly relevant to describing physical scenes. Then, we replace the *subjects and direct objects* of the *verbs and prepositions* (`nsubj, dobj` for verbs and `case` for prepositions in dependency parse trees) with special tokens indicating their dependency roles. For the remaining words, we simply use their POS tags to replace the originals. The four kinds of tokens are illustrated in Table 2–1. Figure 2–1 shows a real example of our normalized sentence representation, where the object pair of interest is <*dog, garden*>.

Table 2–2 Sentence Normalization Example

The	*king*	*opened*	*the*	*door*	*and*	*led*	*the*	***dog***	*into*	*his*	*nice*	***garden***.
DT	open#s	open	DT	open#o	CC	lead	DT	E_1	into	PR	JJ	E_2.

Apart from the normalized tokens of the original sequence, to capture more structural information, we also encode the distances from each token to E_1 and E_2 respectively. Such *position embeddings* (position/distance features) are proposed by [31] with the intuition that information needed to determine the relation between two target nouns normally comes from the words which are close to the target nouns.

We adopt this feature because it can help LSTM keep track of the position of E_1 and E_2, better knowing *where* the two object words are. Then, we leverage LSTM to encode the whole sequence of the tokens of normalized representation plus position embedding. In the meantime, two pretrained *GloVe* word embeddings [23] of the original two physical object words are fed into a hidden dense layer.

Finally, we concatenate both outputs and then use `sigmoid` activation function to obtain the final prediction. We choose to use the popular binary cross-entropy as our loss function, and RMSProp as the optimizer. We apply a dropout rate [32] of 0.5 in the LSTM and embedding layer to prevent overfitting.

2.2 LocatedNear Relation Extraction

The upper part of Figure 2–1 shows the overall workflow of our automatic framework to mine LocatedNear relations from raw text. We first construct a vocabulary of physical objects and generate all candidate instances. For each sentence in the corpus, if a pair of physical objects e_i and e_j appear as nouns in a sentence s, then we apply our sentence-level relation classifier on this instance. The relation classifier yields a probabilistic score s indicating the confidence of the instance in the existence of LocatedNear relation. Finally, all scores of the instances from the corpus are grouped by the object pairs and aggregated, where each object pair is associated with a final score. These mined physical pairs with scores can easily be integrated into existing commonsense knowledge base.

More specifically, for each object pair <e_i, e_j>, we find all the m sentences in our corpus mentioning both objects. We classify the m instances with the sentence-level relation classifier and obtain confidence scores for each instance, then feed them into a heuristic scoring function f to obtain the final aggregated score for

the given object pair. We propose the following 5 choices of f considering accumulation and threshold:

$$f_0 = m \tag{2–1}$$

$$f_1 = \sum_{k=1}^{m} \text{conf}(s_k, e_i, e_j) \tag{2–2}$$

$$f_2 = \frac{1}{m} \sum_{k=1}^{m} \text{conf}(s_k, e_i, e_j) \tag{2–3}$$

$$f_3 = \sum_{k=1}^{m} 1_{\{\text{conf}(s_k, e_i, e_j) > 0.5\}} \tag{2–4}$$

$$f_4 = \frac{1}{m} \sum_{k=1}^{m} 1_{\{\text{conf}(s_k, e_i, e_j) > 0.5\}} \tag{2–5}$$

2.3 Datasets

Our proposed vocabulary of single-word physical objects is constructed by the intersection of all ConceptNet concepts and all entities that belong to "physical object" class in *Wikidata*. We manually filter out some words that have the meaning of an abstract concept, which results in 1,169 physical objects in total.

Afterwards, we utilize a cleaned subset of the Project Gutenberg corpus [33], which contains 3,036 English books written by 142 authors. An assumption here is that sentences in fictions are more likely to describe real life scenes. We sample and investigate the density of LocatedNear relations in Gutenberg with other widely used corpora, namely *Wikipedia*, used by Mintz, Bills, Snow, et al. (2009) and *New York Times* corpus [35]. In the English *Wikipedia* dump, out of all sentences which mentions at least two physical objects, 32.4% turn out to be positive. In the *New York Times* corpus, the percentage of positive sentences is only 25.1%. In contrast, that percentage in the Gutenberg corpus is 55.1%, much higher than the other two corpora, making it a good choice for LocatedNear relation extraction.

From this corpus, we identify 15,193 pairs that co-occur in more than 10 sentences. Among these pairs, we randomly select 500 object pairs and 10 sentences with respect to each pair for annotators to label their commonsense LocatedNear. Each instance is labeled by at least three annotators who are college students and proficient with English. The final truth labels are decided by majority voting. The Cohen's Kappa among the three annotators is 0.711 which suggests substantial agreement [36]. This dataset will be used to train and test models for relation classification. We compare the statistics of our LocatedNear sentence dataset with a few datasets on other well known relations in Table 2–3.

This dataset has almost double the size of those most popular relations in the SemEval task [37], and the sentences in our data set tend to be longer. We randomly choose 4,000 instances as the training set and 1,000 as the test set for evaluating the sentence-level relation classification task.

2.3.1 Commonsense LocatedNear object pairs

We randomly sampled 500 pairs of objects by the number of sentences they appear in. This tends to give us pairs which are more popular.

Table 2–3 Comparison between our LocatedNear dataset and the most popular relations from SemEval 2010 Task 8 dataset for relation classification

Data set	Frequency	Percentage	Words per entry	Chars per word
LocatedNear	2,754	55.1	18.6	4.51
Not LocatedNear	2,246	44.9	19.1	4.32
Cause-Effect	1,331	12.4	17.3	4.71
Component-Whole	1,253	11.7	17.9	4.12
Others	1,864	17.4	17.8	4.34

For the second task, we further ask the annotators to label whether each pair of objects are likely to locate near each other in the real world. Majority votes determine the final truth labels. The inter-annotator agreement here is 0.703 (substantial agreement).

2.4 Evaluation

Table 2–4 Performance of baselines on co-location classification task with ablation. (Acc.=Accuracy, P=Precision, R=Recall, "-" means without certain feature)

	Random	Majority	SVM	SVM(-BW)	SVM(-BPW)	SVM(-BAP)	SVM(-GF)
Acc.	0.500	0.551	0.584	0.577	0.556	0.563	**0.605**
P	0.551	0.551	0.606	0.579	0.567	0.573	**0.616**
R	0.500	1.000	0.702	0.675	0.681	**0.811**	0.751
F1	0.524	0.710	0.650	0.623	0.619	0.672	**0.677**
	SVM(-SDP)	SVM(-SS)	DRNN	LSTM+Word	LSTM+POS	LSTM+Norm	
Acc.	0.579	0.584	0.635	0.637	0.641	**0.653**	
P	0.597	0.605	**0.658**	0.635	0.650	0.654	
R	0.728	0.708	0.702	**0.800**	0.751	0.784	
F1	0.656	0.652	0.679	0.708	0.697	**0.713**	

In this section, we first present our evaluation of our proposed methods and the state-of-the-art general relation classification model on the first task. Then, we evaluate the quality of the new LocatedNear triples we extracted.

2.4.1 Sentence-level LocatedNear Relation Classification

We evaluate the proposed methods against the state-of-the-art general domain relation classification model (DRNN) [38]. The results are shown in Table 2–4. For feature-based SVM, we do feature ablation on each of the 6 feature types. For LSTM-based model, we experiment on variants of input sequence of original sentence: "LSTM+Word" uses the original words as the input tokens; "LSTM+POS" uses only POS tags as the input tokens; "LSTM+Norm" uses the tokens of sequence after sentence normalization. Besides, we add two naive baselines: "Random" baseline method classifies the instances into two classes with equal

Table 2–5 Ranking results of scoring functions.

f	MAP	P@50	P@100	P@200	P@300
f_0	0.42	0.40	0.44	0.42	0.38
f_1	0.58	**0.70**	0.60	0.53	**0.44**
f_2	0.48	0.56	0.52	0.49	0.42
f_3	**0.59**	0.68	**0.63**	**0.55**	**0.44**
f_4	0.56	0.40	0.48	0.50	0.42

probability. "Majority" baseline method considers all the instances to be positive.

From the results, we find that the SVM model without the Global Features performs best, which indicates that bag-of-word features benefit more in shortest dependency paths than on the whole sentence. Also, we notice that DRNN performs best (0.658) on precision but not significantly higher than LSTM+Norm (0.654). The experiment shows that LSTM+Word enjoys the highest recall score, while LSTM+Norm is the best one in terms of the overall performance. One reason is that the normalization representation reduces the vocabulary of input sequences, while also preserving important syntactical and semantic information. Another reason is that the LocatedNear relation are described in sentences decorated with prepositions/adverbs. These words are usually descendants of the object word in the dependency tree, outside of the shortest dependency paths. Thus, DRNN cannot capture the information from the words belonging to the descendants of the two object words in the tree, but this information is well captured by LSTM+Norm.

2.4.2 LocatedNear Relation Extraction

Once we have obtained the probability score for each instance using LSTM+Norm, we can extract LocatedNear relation using the scoring function f. We compare the performance of 5 different heuristic choices of f, by quantitative results. We rank 500 commonsense LocatedNear object pairs described in Section 2.2. Table 2–5 shows the ranking results using *Mean Average Precision* (MAP) and *Precision* at K as the metrics. Accumulative scores (f_1 and f_3) generally do better. Thus, we choose $f = f_3$ with a MAP score of 0.59 as the scoring function.

Table 2–6 Top object pairs returned by best performing scoring function f_3

(door, room)	(boy, girl)	(cup, tea)
(ship, sea)	(house, garden)	(arm, leg)
(fire, wood)	(house, fire)	(horse, saddle)
(fire, smoke)	(door, hall)	(door, street)
(book, table)	(fruit, tree)	(table, chair)

Qualitatively, we show 15 object pairs with some of the highest f_3 scores in Table 2–6. Setting a threshold of 40.0 for f_3, which is the minimum non-zero f_3 score for all true object pairs in the LocatedNear object pairs data set (500 pairs), we obtain a total of 2,067 LocatedNear relations, with a precision of 68% by human inspection.

2.5 Related Work

Classifying relations between entities in a certain sentence plays a key role in NLP applications and thus has been a hot research topic recently. Feature-based methods [37] and neural network techniques [39, 40] are most common. Xu, Mou, Li, et al. (2015) introduce multi-channel SDP-based LSTM model to classify relations incorporating several different kinds of information of a sentence, which performed best on SemEval-2010 Task 8 and is one of our baseline methods.

The most related work to ours is the extraction of visual commonsense knowledge by Yatskar, Ordonez, Farhadi (2016). This work learns the textual representation of seven types of fine-grained visual relations using textual caption for the image in MS-COCO dataset [41], such as "touches", "above" and "disconnected from" by jointly modeling the relative position of the 80 kinds of objects in 300,000 images and the textual caption for the image in MS-COCO dataset[41][41]. The authors generalized their extracted knowledge using WordNet. Their resource are not scalable for its expensive human labor, and we propose a framework to use large text which is scalable and involves more real world description. Another important related work is from Li, Taheri, Tu, et al. (2016), which enriches several popular relations in *ConceptNet* with little textual information from real large corpora. However, LocatedNear relation was not studied in this work, while this relation is extremely scarce in *ConceptNet* and has its own distinctiveness.

2.6 Conclusion

In this paper, we present a novel study on enriching LocatedNear relationship from textual corpora. Based on our two newly-collected benchmark datasets, we propose several methods to solve the sentence-level relation classification problem. We show that existing methods do not work as well on this task and discovered that LSTM-based model does not have significant edge over simpler feature-based model. Whereas, our multi-level sentence normalization turns out to be useful.

Future directions include: 1) better leveraging distant supervision to reduce human efforts, 2) incorporating knowledge graph embedding techniques, 3) applying the LocatedNear knowledge into downstream applications in computer vision and natural language processing.

Chapter 3 Text-Enhanced Common-Sense Knowledge Graph Representation Learning

3.1 Approaches

There are several works using textual information to help KG representation learning. The neural tensor network (NTN) model [21] represents each relation with a bilinear operator and represents each node by averaging the word vectors in the node, allowing the sharing of textual information located in similar nodes. DistMult [12] is a simplified bilinear model, where each relation is represented as a diagonal matrix and each node is represented as either a single vector or an average of word vectors. [20] proposes a joint method by aligning entity with entity name and its WikiPedia anchor, which contains knowledge embedding model, word embedding model and corresponding alignment model as the joint part. [14] extends this model by aligning entity to the corresponding entity description. Xie(2016) employ a CNN to encode the entity description as textual representation of entity, and jointly learn knowledge graph embedding by considering separate energy functions based on structure-based representation and description-based representation. These methods separate the objective functions into two energy functions of structure-based and textual-based representations. To utilize both representations, they need further estimate an optimum weight coefficients to combine them together in the specific tasks.

[16] firstly integrate the representations of structure and text into a unified representation through a static gating strategy based on the textual description of nodes. The intuition is similar. In this paper, we explore several simpler integration methods with fewer parameters based on the textual information conveyed by the nodes. And we obtain better results.

In this section, we briefly discuss TransE, which will be used as our baseline. Next we present our model.

TranE

For each triplet (h, r, t) in a given knowledge graph, TransE defines the following dissimilarity score function:

$$s(h, r, t) = ||\mathbf{h} + \mathbf{r} - \mathbf{t}||$$

where $\mathbf{h}, \mathbf{t}, \mathbf{r} \in \mathbb{R}^d$ are d-dimensional vector representations for the left node h, the right node t, and relation r respectively. TransE regards $\mathbf{r}$ as a translation operator between $\mathbf{h}$ and $\mathbf{t}$, such that $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$. It aims to assign lower scores to valid triplets over invalid ones.

Our Model

In our approach, we learn to represent each concept node with a low-dimensional vector representation that captures both node-level textual information as well as the structural information from the knowledge graph.

We consider two simple mechanisms for combining such two types of information, namely linear combination and concatenation.

3.1.0.0.1 Linear Combination: In this approach, we define the node representations as:

$$\mathbf{h}' = \mathbf{h}_{\text{graph}} + \lambda \mathbf{h}_{\text{text}},$$
$$\mathbf{t}' = \mathbf{t}_{\text{graph}} + \lambda \mathbf{t}_{\text{text}},$$

where $\mathbf{h}_{\text{graph}}, \mathbf{t}_{\text{graph}} \in \mathbb{R}^d$ are vector representations that capture structural information from the knowledge graph, λ is a hyper-parameter, and $\mathbf{h}_{\text{text}}, \mathbf{t}_{\text{text}} \in \mathbb{R}^d$ are vectors that capture the textual information conveyed by the left concept node h and right concept node t respectively. We define the textual feature vector for each node as the composition of vectors of the sequence of words that appear in each node:

$$\mathbf{h}_{\text{text}} = f(\mathbf{w}_h^{(1)}, \ldots, \mathbf{w}_h^{(|h|)})$$
$$\mathbf{t}_{\text{text}} = f(\mathbf{w}_t^{(1)}, \ldots, \mathbf{w}_t^{(|t|)})$$

where $\mathbf{w}_h^{(k)}$ and $\mathbf{w}_t^{(k)}$ are the representations for words at position k of the textual descriptions of the left concept node h with the length (i.e., number of words) of $|h|$ and right concept node t with the length of $|t|$ respectively. We consider two possible composition functions for f: 1) word vector averaging (AVG) and 2) using a long short-term memory recurrent neural network (LSTM).

3.1.0.0.2 Concatenation: The node representations with the concatenation approach are defined as:

$$\mathbf{h}' = g(\mathbf{W}[\mathbf{h}_{\text{graph}}; \mathbf{h}_{\text{text}}] + \mathbf{b})$$
$$\mathbf{t}' = g(\mathbf{W}[\mathbf{t}_{\text{graph}}; \mathbf{t}_{\text{text}}] + \mathbf{b})$$

where $[\cdot\,;\,\cdot]$ is the concatenation operation that concatenates two vectors, g is a non-linear activation function, $\mathbf{W}$ is a parameter matrix and $\mathbf{b}$ is the bias vector.

Objective Function

We still use a single vector $\mathbf{r} \in \mathbb{R}^d$ to represent the relation r. The score function is defined as:

$$s(h, r, t) = ||\mathbf{h}' + \mathbf{r} - \mathbf{t}'||_2^2$$

We train our model such that it assigns lower scores to valid triplets and higher scores to invalid ones. We aim to achieve this by minimizing the following margin-based ranking loss:

$$\mathcal{L} = \sum_{(h,r,t) \in S} \sum_{(\hat{h},r,\hat{t}) \in \hat{S}} [s(h, r, t) + \gamma - s(\hat{h}, r, \hat{t})]_+$$

where $[x]_+ = \max(0, x)$, and γ is the margin. The set S consists of all valid triplets in the training set, and $\hat{S}$ is the set of invalid triplets generated by sampling from the following set:

$$\{(\hat{h}, r, t) | \hat{h} \in C, (\hat{h}, r, t) \notin S\}$$
$$\cup \{(h, r, \hat{t}) | \hat{t} \in C, (h, r, \hat{t}) \notin S\}$$

where C is the set of concept nodes from the existing commonsense knowledge graph.

Table 3–1 Dataset statistics

#Dataset	#Relation	#Nodes	#Train	#Dev	#Test
FB15k	1345	14951	483142	50000	59071
WN18	18	40493	141442	5000	5000
CN22	22	17,216	86,991	10,861	10,899

Table 3–2 Evaluation results on link prediction on FB15k and WN18

Method	FB15K				WN18			
	Mean Rank		Hits@10		Mean Rank		Hits@10	
	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter
TransE [9]	210	119	48.5	66.1	263	251	75.4	89.2
TransH [10]	212	87	45.7	64.4	318	303	75.4	86.7
DistMult [12]	-	-	-	57.7	-	-		**94.2**
DESP [14]	167	39	51.7	77.3	-	-	-	-
DKRL [15]	181	91	49.6	67.4	-	-	-	-
TEKE [45]	233	79	43.5	67.6	240	127	80.0	93.8
SSP [46]	163	82	**57.2**	79.0	168	156	**81.2**	93.2
Jointly [16]	167	77	52.9	75.5	117	95	79.5	91.6
JointE+SATT [47]	-	-	-	79.3	-	-	-	-
Our Model (concat, AVG)	199	57	47.8	70.9	131	116	75.2	87.4
Our Model (linear, AVG)	202	**52**	50.8	**80.9**	**105**	**91**	79.6	93.8

3.2 Experiments

3.2.1 Dataset

We use two public datasets: WN18 [22] and FB15K [9]. We also created another evaluation dataset created based on the latest ConceptNet [42][1]. Following a similar rule used for constructing WN18 [22], we filter out relation types appearing in less than 1,000 triplets as well as concepts appearing in less than 10 triplets. Finally we obtain a sub-graph of ConceptNet consisting of 17,216 nodes and 22 relation types, which we name as CN22 and release at #URL#. We randomly split 80% of CN22 for training, 10% for development and 10% for evaluation. Table 3–1 shows the statistics of the resulting datasets.

3.2.2 Implementation Details

We use AdaGrad [43] as the optimization method with the initial learning rate set as 0.1. We empirically set the margin γ to 1, the dimension d to 200 for node and relation representations. The Glove 200-dimensional word vectors trained on Wikipedia 2014 and Gigaword 5 [44][2] are used in this work for generating the textual representations of nodes.[3] We set the maximum number of training iterations as 1,000 and use the development set to determine when to stop training. We also tune λ with development set (optimal $\lambda = 1$), and use tanh as the non-linear activation function.

[1] https://github.com/commonsense/conceptnet5

[2] https://nlp.stanford.edu/projects/glove/

[3] In fact, we can also learn domain or corpus specific word vectors, which we leave as a future work.

Table 3–3 Evaluation results on link prediction on CN22

Method	Mean Rank				Hits@10(%)			
	Head		Tail		Head		Tail	
	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter
TransE	2109	2087	1535	1529	8.4	12.5	14.2	18.0
DistMult	3679	3659	3109	3094	5.2	7.6	9.4	11.8
DistMult (AVG)	1983	1963	1914	1908	8.4	13.4	9.4	14.6
DistMult (LSTM)	1568	1550	1248	1211	6.9	7.4	11.8	12.5
Our Model (concat, AVG)	1135	1115	794	787	11.9	15.6	18.5	21.4
Our Model (concat, LSTM)	1320	1299	914	908	10.7	13.5	16.1	19.2
Our Model (linear, AVG)	**955**	**935**	**658**	**651**	**13.2**	**17.5**	**19.9**	**23.3**
Our Model (linear, LSTM)	1005	984	704	697	12.2	15.5	18.5	21.8

3.2.3 Link Prediction Task

This task aims to answer queries of the following types: $(?, r, t)$ or $(h, r, ?)$. In other words, it tries to predict a missing node of a triplet such as predicting the left concept node h given (r, t) or predicting the right concept node t given (h, r). To do so, for each test query, following [9], we replace the missing node by each possible concept node that appeared in the existing knowledge graph to measure the score of the constructed triplet: $s(h, r, t)$. We next sort these scores, based on which we return the rank of the desired node. We report such rank information under "Raw". As sometimes there may be multiple nodes that satisfy the same test query, when examining the rank of a specific node, we could also remove all the other nodes that also satisfy the query when calculating the rank. Such results are reported under "Filter". Following previous approaches [9, 10], we consider two standard metrics for reporting the rank information: the average rank of the expected node (Mean Rank), and the proportion of test triplets that appear within the top-10 list (Hits@10). A lower Mean Rank or a higher Hits@10 score indicates a better performance.

3.2.4 Results

Considering two combination methods and two composition methods, there are four proposed models to explore. The link prediction results on FB15k and WN18 are shown in Table 3–2. We compare our methods against basic translation models as well as various variants considering textual information about nodes. Besides, we compare against DistMult, which also consider the textual information conveyed by the nodes. For our models, we can observe that models with word vectors averaging (AVG) method perform better than those with LSTM. We believe one possible reason is that the lengths of words in the nodes are relatively short in our CN22 dataset, with the maximum length being only 6. While LSTM may be more suitable for capturing long distance information in text, AVG as a simple method might be sufficient for this dataset. Perhaps due to a similar reason, we can also observe that the simple linear combination method performs better than the more sophisticated concatenation method. Overall, our model with a linear combination approach for integrating both textual and structural features, where the textual features are constructed with word vector averaging obtains the best performance.

Table 3–3 shows the evaluation results on CN22. As we cannot obtain external textual corpus about nodes, we just comapre against basic TransE and DistMult. As highlighted in [12], the difference between

DistMult and TransE is the choice of the composition operation of two node vectors. DistMult uses a multiplication operation while TransE uses an additive operation. ConceptNet is a smaller and sparser knowledge graph compared to WordNet and Freebase. It can be observed that TransE, which regards relations as linear translation operators, seems to be more suitable than DistMult for learning representations for such a knowledge graph. When textual information in the nodes is incorporated into the representation learning process of DistMult model, a better performance can be obtained. Such results show the importance of textual information for learning such a knowledge graph. Our models that capture both textual information in the nodes and the structural information from the graph generally perform better than TransE and DistMult models.

3.3 Conclusion

In this work, we explore several simple and general models for text-enhanced knowledge graph representation learning. Through experiments, we observe that our models which combine textual features conveyed by the nodes and structural features from graph together perform well compared to other models. We also empirically found that the simple approach that makes use of word vector averaging to construct textual representations while using a linear combination approach to integrate both textual and structural information can yield better results as compared to existing baseline approaches.

Chapter 4 Multi-channel BiLSTM-CRF Model for Recognizing Novel Entity in Social Media

Named entity recognition (NER) is one of the first and most important steps in Information Extraction pipelines. Generally, it is to identify mentions of entities (persons, locations, organizations, etc.) within unstructured text. However, the diverse and noisy nature of user-generated content as well as the emerging entities with novel surface forms make NER in social media messages more challenging.

The first challenge brought by user-generated content is its unique characteristics: short, noisy and informal. For instance, tweets are typically short since the number of characters is restricted to 140 and people indeed tend to pose short messages even in social media without such restrictions, such as YouTube comments and Reddit. [1] Hence, the contextual information in a sentence is very limited. Apart from that, the use of colloquial language makes it more difficult for existing NER approaches to be reused, which mainly focus on a general domain and formal text [48, 49].

Another challenge of NER in noisy text is the fact that there are large amounts of emerging named entities and rare surface forms among the user-generated text, which tend to be tougher to detect [50] and recall thus is a significant problem [49]. By way of example, the surface form "*kktny*", in the tweet "so.. *kktny* in 30 mins?", actually refers to a new TV series called "*Kourtney and Kim Take New York*", which even human experts found hard to recognize. Additionally, it is quite often that netizens mention entities using rare morphs as surface forms. For example, "*black mamba*", the name for a venomous snake, is actually a morph that Kobe Bryant created for himself for his aggressiveness in playing basketball games [51]. Such morphs and rare surface forms are also very difficult to detect and classify.

The goal of this paper is to present our system participating in the *Novel and Emerging Named Entity Recognition* shared task at the EMNLP 2017 Workshop on Noisy User-generated Text (W-NUT 2017), which aims for NER in such noisy user-generated text. We investigate a multi-channel BiLSTM-CRF neural network model in our participating system, which is described in Section 4.2. The details of our implementation are in presented in Section 4.3, where we also present some conclusion from our experiments.

4.1 Problem Definition

The NER is a classic sequence labeling problem, in which we are given a sentence, in the form of a sequence of tokens $\mathbf{w} = (w_1, w_2, ..., w_n)$, and we are required to output a sequence of token labels $\mathbf{y} = (y_1, y_2, ..., y_n)$. In this specific task, we use the standard BIO2 annotation, and each named entity chunk are classified into 6 categories, namely Person, Location (including GPE, facility), Corporation, Consumer good (tangible goods, or well-defined services), Creative work (song, movie, book, and so on) and Group (subsuming music band, sports team, and non-corporate organizations).

[1] The average length of the sentences in this shared task is about 20 tokens per sentence.

4.2 Approach

In this section, we will first introduce the overview of our proposed model and then present each part of the model in detail.

4.2.1 Overview

Figure 4–1 shows the overall structure of our proposed model, instead of solely using the original pretrained word embeddings as the final word representations, we construct a comprehensive word representation for each word in the input sentence. This comprehensive word representations contain the character-level sub-word information, the original pretrained word embeddings and multiple syntactical features. Then, we feed them into a Bidirectional LSTM layer, and thus we have a hidden state for each word. The hidden states are considered as the feature vectors of the words by the final CRF layer, from which we can decode the final predicted tag sequence for the input sentence.

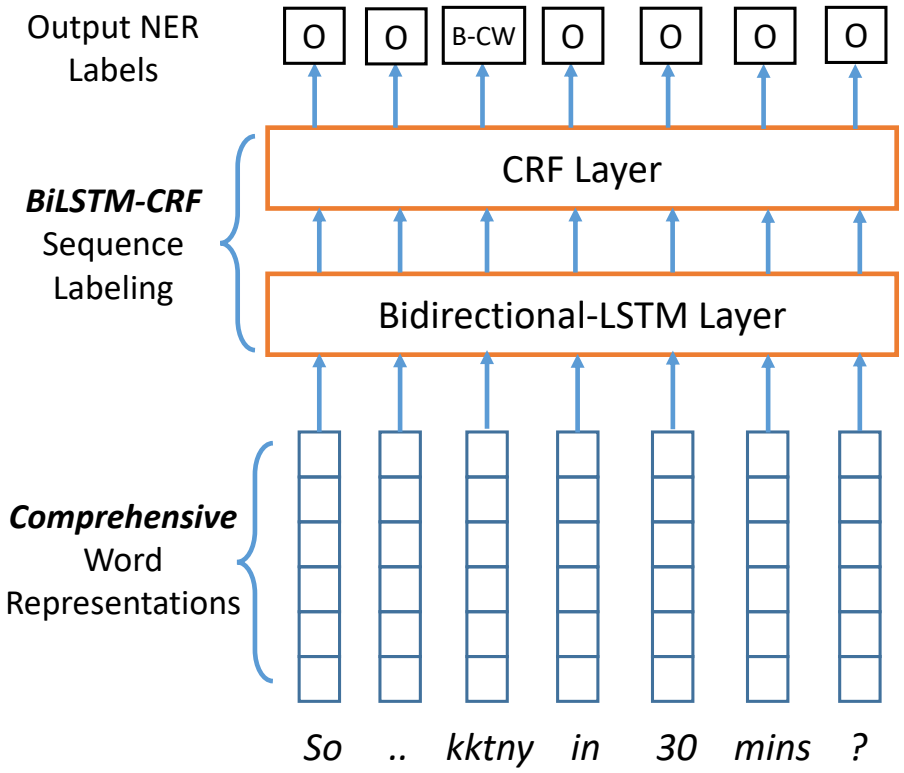

Figure 4–1 Overview of our approach.

4.2.2 Comprehensive Word Representations

In this subsection, we present our proposed comprehensive word representations. We first build character-level word representations from the embeddings of every character in each word using a bidirectional LSTM.

Then we further incorporate the final word representation with the embedding of the syntactical information of each token, such as the part-of-speech tag, the dependency role, the word position in the sentence and the head position. Finally, we combine the original word embeddings with the above two parts to obtain the final comprehensive word representations.

4.2.2.1 Character-level Word Representations

In noisy user-generated text analysis, sub-word (character-level) information is much more important than that in normal text analysis for two main reasons: 1) People are more likely to use novel abbreviations and morphs to mention entities, which are often out of vocabulary and only occur a few times. Thus, solely using the original word-level word embedding as features to represent words is not adequate to capture the characteristics of such mentions. 2) Another reason why we have to pay more attention to character-level word representation for noisy text is that it is can capture the orthographic or morphological information of both formal words and Internet slang.

There are two main network structures to make use of character embeddings: one is CNN [52] and the other is BiLSTM[53]. BiLSTM turns to be better in our experiment on development dataset. Thus, we follow Lample et al. (2016) to build a BiLSTM network to encode the characters in each token as Figure 4–2 shows. We finally concatenate the forward embedding and backward embedding to the final character-level word representation.

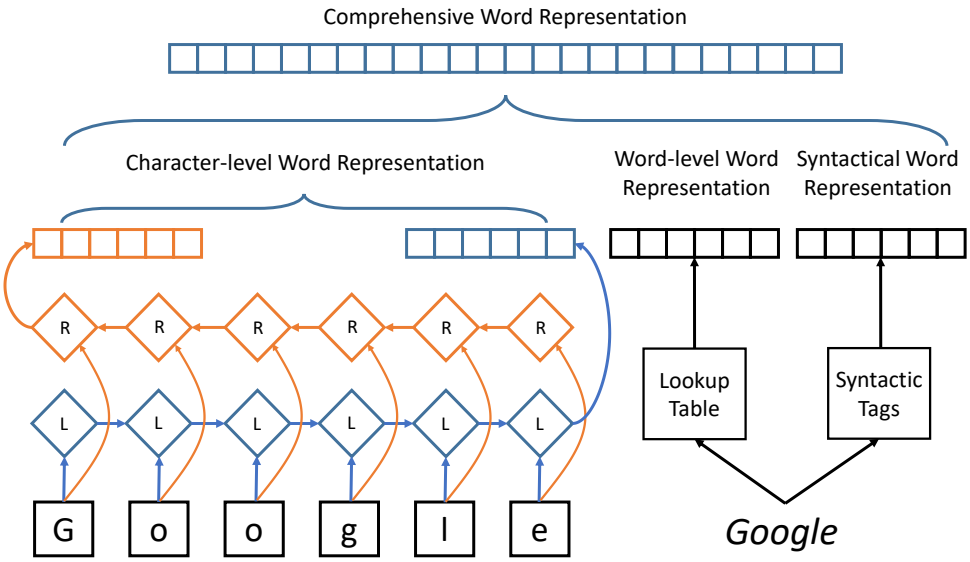

Figure 4–2 Illustration of comprehensive word representations.

4.2.2.2 Syntactical Word Representations

We argue that the syntactical information, such as POS tags and dependency roles, should also be explicitly considered as contextual features of each token in the sentence.

TweetNLP and TweeboParser [54, 55] are two popular software to generate such syntactical tags for each token given a tweet. Given the nature of the noisy tweet text, a new set of POS tags and dependency trees are used in the tool, called Tweebank [56]. See Table 4–1 for an example POS tagging. Since a tweet often contains more than one utterance, the output of TweeboParser will often be a multi-rooted graph over the tweet.

Word position embedding are included as well as it is widely used in other similar tasks, like relation classification [57]. Also, head position embeddings are taken into account while calculating these embedding vectors to further enrich the dependency information. It tries to exclude these tokens from the parse tree, resulting a head index of -1.

After calculating all 4 types of embedding vectors (POS tags, dependency roles, word positions, head positions) for every tokens, we concatenate them to form a syntactical word representation.

Table 4–1 Example of POS tagging for tweets.

Token	so	..	kktny	in	30	mins	?
POS	R	,	N	P	$	N	,
Position	1	2	3	4	5	6	7
Head	0	-1	0	3	6	4	-1

4.2.2.3 Combination with Word-level Word Representations

After obtaining the above two additional word representations, we combine them with the original word-level word representations, which are just traditional word embeddings.

To sum up, our comprehensive word representations are the concatenation of three parts: 1) character-level word representations, 2) syntactical word representation and 3) original pretrained word embeddings.

4.2.3 BiLSTM Layer

LSTM based networks are proven to be effective in sequence labeling problem for they have access to both past and the future contexts. Whereas, hidden states in unidirectional LSTMs only takes information from the past, which may be adequate to classify the sentiment is a shortcoming for labeling each token. Bidirectional LSTMs enable the hidden states to capture both historical and future context information and then to label a token.

Mathematically, the input of this BiLSTM layer is a sequence of comprehensive word representations (vectors) for the tokens of the input sentence, denoted as $(\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_n})$. The output of this BiLSTM layer is a sequence of the hidden states for each input word vectors, denoted as $(\mathbf{h_1}, \mathbf{h_2}, ..., \mathbf{h_n})$. Each final hidden state is the concatenation of the forward $\overleftarrow{\mathbf{h_i}}$ and backward $\overrightarrow{\mathbf{h_i}}$ hidden states. We know that

$$\overleftarrow{\mathbf{h_i}} = \text{lstm}(\mathbf{x_i}, \overleftarrow{\mathbf{h_{i-1}}}) \, , \, \overrightarrow{\mathbf{h_i}} = \text{lstm}(\mathbf{x_i}, \overrightarrow{\mathbf{h_{i+1}}})$$

$$\mathbf{h_i} = \left[\overleftarrow{\mathbf{h_i}} ; \overrightarrow{\mathbf{h_i}} \right]$$

4.2.4 CRF Layer

It is almost always beneficial to consider the correlations between the current label and neighboring labels since there are many syntactical constrains in natural language sentences. For example, I-PERSON will never follow a B-GROUP. If we simply feed the above mentioned hidden states independently to a Softmax layer to predict the labels, then such constrains will not be more likely to be broken. Linear-chain Conditional Random Field is the most popular way to control the structure prediction and its basic idea is to use a series of potential function to approximate the conditional probability of the output label sequence given the input word sequence.

Formally, we take the above sequence of hidden states $\mathbf{h} = (\mathbf{h_1}, \mathbf{h_2}, ..., \mathbf{h_n})$ as our input to the CRF layer, and its output is our final prediction label sequence $\mathbf{y} = (y_1, y_2, ..., y_n)$, where y_i is in the set of all possible labels. We denote $\mathcal{Y}(\mathbf{h})$ as the set of all possible label sequences. Then we derive the conditional probability of the output sequence given the input hidden state sequence is

$$p(\mathbf{y}|\mathbf{h}; \mathbf{W}, \mathbf{b}) = \frac{\prod_{i=1}^{n} \exp(\mathbf{W}^T_{y_{i-1}, y_i} \mathbf{h} + \mathbf{b}_{y_{i-1}, y_i})}{\sum_{\mathbf{y}' \in \mathcal{Y}(\mathbf{h})} \prod_{i=1}^{n} \exp(\mathbf{W}^T_{y'_{i-1}, y'_i} \mathbf{h} + \mathbf{b}_{y'_{i-1}, y'_i})}$$

, where $\mathbf{W}$ and $\mathbf{b}$ are the two weight matrices and the subscription indicates that we extract the weight vector for the given label pair (y_{i-1}, y_i).

To train the CRF layer, we use the classic maximum conditional likelihood estimation to train our model. The final log-likelihood with respect to the weight matrices is

$$L(\mathbf{W}, \mathbf{b}) = \sum_{(\mathbf{h_i}, \mathbf{y_i})} \log p(\mathbf{y_i}|\mathbf{h_i}; \mathbf{W}, \mathbf{b})$$

Finally, we adopt the Viterbi algorithm for training the CRF layer and the decoding the optimal output sequence $\mathbf{y}^*$.

4.3 Experiments

In this section, we discuss the implementation details of our system such as hyper parameter tuning and the initialization of our model parameters. [1]

[1]The detailed description of the evaluation metric and the dataset are shown in `http://noisy-text.github.io/2017/emerging-rare-entities.html`

4.3.1 Parameter Initialization

For word-level word representation (i.e. the lookup table), we utilize the pretrained word embeddings[1] from GloVe[44]. For all out-of-vocabulary words, we assign their embeddings by randomly sampling from range $\left[-\sqrt{\frac{3}{\text{dim}}}, +\sqrt{\frac{3}{\text{dim}}}\right]$, where *dim* is the dimension of word embeddings, suggested by He et al.(2015). The random initialization of character embeddings are in the same way. We randomly initialize the weight matrices $\mathbf{W}$ and $\mathbf{b}$ with uniform samples from $\left[-\sqrt{\frac{6}{r+c}}, +\sqrt{\frac{6}{r+c}}\right]$, r and c are the number of the rows and columns, following Glorot and Bengio(2010). The weight matrices in LSTM are initialized in the same work while all LSTM hidden states are initialized to be zero except for the bias for the forget gate is initialized to be 1.0 , following Jozefowicz et al.(2015).

4.3.2 Hyper Parameter Tuning

We tuned the dimension of word-level embeddings from {50, **100**, 200}, character embeddings from {10, **25**, 50}, character BiLSTM hidden states (i.e. the character level word representation) from {20, **50**, 100}. We finally choose the bold ones. The dimension of part-of-speech tags, dependecny roles, word positions and head positions are all 5.

As for learning method, we compare the traditional SGD and Adam [61]. We found that Adam performs always better than SGD, and we tune the learning rate form {1e-2,**1e-3**,1e-4}.

4.3.3 Results

To evaluate the effectiveness of each feature in our model, we do the feature ablation experiments and the results are shown in Table 4–2.

Table 4–2 Feature Ablation

Features	F1 (entity)	F1 (surface form)
Word	37.16	34.15
Char(LSTM)+Word	38.24	37.21
POS+Char(LSTM)+Word	40.01	37.57
Syntactical+Char(CNN)+Word	40.12	37.52
Syntactical+Char(LSTM)+Word	**40.42**	**37.62**

In comparison with other participants, the results are shown in Table 4–3.

Table 4–3 Result comparison

Team	F1 (entity)	F1 (surface form)
Drexel-CCI	26.30	25.26
MIC-CIS	37.06	34.25
FLYTXT	38.35	36.31
Arcada	39.98	37.77
Ours	**40.42**	**37.62**
SpinningBytes	40.78	39.33
UH-RiTUAL	**41.86**	**40.24**

[1] `http://nlp.stanford.edu/data/glove.twitter.27B.zip`

4.4 Conclusion

In this paper, we present a novel multi-channel BiLSTM-CRF model for emerging named entity recognition in social media messages. We find that BiLST-CRF architecture with our proposed comprehensive word representations built from multiple information are effective to overcome the noisy and short nature of social media messages.

Chapter 5 Mining Cross-Cultural Differences and Similarities in Social Media

5.1 The *SocVec* Framework

In this section, we first discuss the intuition behind our model, the concept of "social words" and our notations. Then, we present the overall workflow of our approach. We finally describe the *SocVec* framework in detail.

5.1.1 Problem Statement

We choose (English, Chinese) to be the target language pair throughout this paper for the salient cross-cultural differences between the east and the west[1]. Given an English term W and a Chinese term U, the core research question is how to compute a similarity score, $ccsim(W, U)$, to represent the *cross-cultural similarities* between them.

We cannot directly calculate the similarity between the monolingual word vectors of W and U, because they are trained separately and the semantics of dimension are not aligned. Thus, the challenge is to devise a way to compute similarities across two different vector spaces while retaining their respective cultural characteristics.

A very intuitive solution is to firstly translate the Chinese term U to its English counterpart U' through a Chinese-English bilingual lexicon, and then regard $ccsim(W, U)$ as the (cosine) similarity between W and U' with their monolingual word embeddings. However, this solution is not promising in some common cases for three reasons:

1. if U is an OOV (Out of Vocabulary) term, e.g., a novel slang term, then there is probably no translation U' in bilingual lexicons.
2. if W and U are names referring to the same named entity, then we have $U' = W$. Therefore, $ccsim(W, U)$ is just the similarity between W and itself, and we cannot capture any cross-cultural differences with this method.
3. this approach does not explicitly preserve the cultural and social contexts of the terms.

To overcome the above problems, our intuition is to project both English and Chinese word vectors into a single third space, known as *SocVec*, and the projection is supposed to purposely carry cultural features of terms.

5.1.2 Social Words and Our Notations

Some research in psychology and sociology [62, 63] show that culture can be highly related to emotions and opinions people express in their discussions. As suggested by [64], we thus define the concept of

[1]Nevertheless, the techniques are language independent and thus can be utilized for any language pairs so long as the necessary resources outlined in Section 5.1.3 are available.

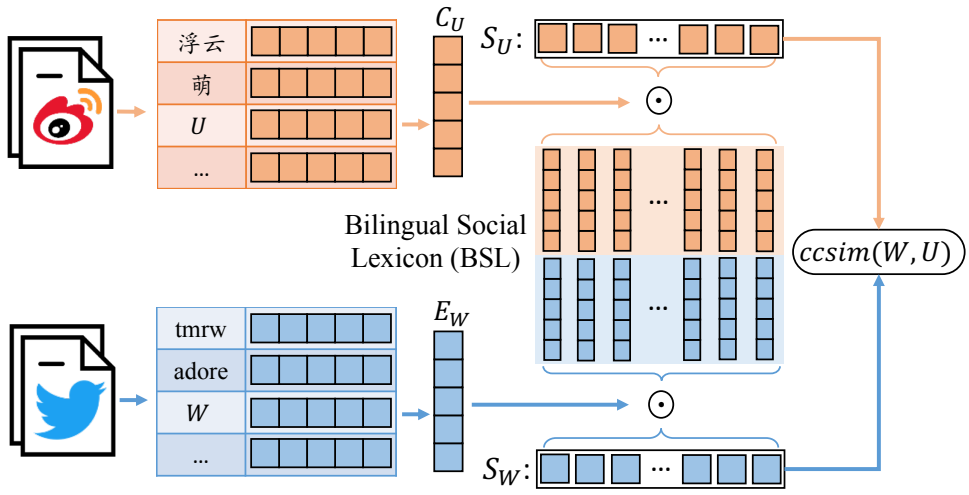

Figure 5–1 Workflow for computing the cross-cultural similarity between an English word W and a Chinese word U, denoted by $ccsim(W, U)$

"**social word**" as the words directly reflecting opinion, sentiment, cognition and other human psychological processes[1], which are important to capturing cultural and social characteristics. Both [65] and [66] find such *social words* are most effective culture/socio-linguistic features in identifying cross-cultural differences.

We use these notations throughout the paper: *CnVec* and *EnVec* denote the Chinese and English word vector space, respectively; *CSV* and *ESV* denote the Chinese and English social word vocab; *BL* means Bilingual Lexicon, and *BSL* is short for Bilingual Social Lexicon; finally, we use $\mathbf{E_x}$, $\mathbf{C_x}$ and $\mathbf{S_x}$ to denote the word vectors of the word x in *EnVec*, *CnVec* and *SocVec* spaces respectively.

5.1.3 Overall Workflow

Figure 5–1 shows the workflow of our framework to construct the *SocVec* and compute $ccsim(W, U)$. Our proposed *SocVec* model attacks the problem with the help of three low-cost external resources: (i) an English corpus and a Chinese corpus from social media; (ii) an English-to-Chinese bilingual lexicon (*BL*); (iii) an English social word vocabulary (*ESV*) and a Chinese one (*CSV*).

We train English and Chinese word embeddings (*EnVec* and *CnVec*) on the English and Chinese social media corpus respectively. Then, we build a *BSL* from the *CSV*, *ESV* and *BL* (see Section 5.1.4). The *BSL* further maps the previously incompatible *EnVec* and *CnVec* into a single common vector space *SocVec*, where two new vectors, S_W for W and S_U for U, are finally comparable.

5.1.4 Building the BSL

The process of building the *BSL* is illustrated in Figure 5–2. We first extract our bilingual lexicon (*BL*), where confidence score w_i represents the probability distribution on the multiple translations for each word.

[1]Example social words in English include *fawn, inept, tremendous, gratitude, terror, terrific, loving, traumatic*, etc. We discuss the sources of such social words in Section 5.2.

Afterwards, we use BL to translate each social word in the *ESV* to a set of Chinese words and then filter out all the words that are not in the *CSV*. Now, we have a set of Chinese social words for each English social word, which is denoted by a "translation set". The final step is to generate a Chinese "pseudo-word" for each English social word using their corresponding translation sets. A "pseudo-word" can be either a real word that is the most representative word in the translation set, or an imaginary word whose vector is a certain combination of the vectors of the words in the translation set.

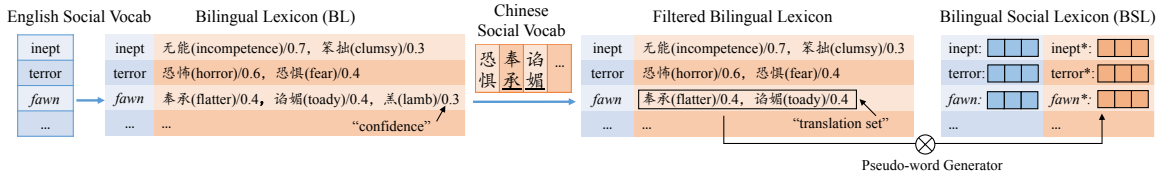

Figure 5–2 Generating an entry in the BSL for "*fawn*" and its pseudo-word "*fawn**"

For example, in Figure 5–2, the English social word "*fawn*" has three Chinese translations in the bilingual lexicon, but only two of them (underlined) are in the CSV. Thus, we only keep these two in the translation set in the filtered bilingual lexicon. The pseudo-word generator takes the word vectors of the two words (in the black box), namely 奉承 (flatter) and 谄媚 (toady), as input, and generates the pseudo-word vector denoted by "*fawn**". Note that the direction of building *BSL* can also be from Chinese to English, in the same manner. However, we find that the current direction gives better results due to the better translation quality of our *BL* in this direction.

Given an English social word, we denote $\mathbf{t_i}$ as the i^{th} Chinese word of its translation set consisting of N social words. We design four intuitive types of pseudo-word generator as follows, which are tested in the experiments:

(1) Max. Maximum of the values in each dimension, assuming dimensionality is K:

$$\text{Pseudo}(\mathbf{C_{t_1}}, ..., \mathbf{C_{t_N}}) = \begin{bmatrix} max(C^{(1)}_{t_1}, ..., C^{(1)}_{t_N}) \\ \vdots \\ max(C^{(K)}_{t_1}, ..., C^{(K)}_{t_N}) \end{bmatrix}^{\mathrm{T}}$$

(2) Avg. Average of the values in every dimension:

$$\text{Pseudo}(\mathbf{C_{t_1}}, ..., \mathbf{C_{t_N}}) = \frac{1}{N}\sum_{i}^{N} \mathbf{C_{t_i}}$$

(3) WAvg. Weighted average value of every dimension with respect to the translation confidence:

$$\text{Pseudo}(\mathbf{C_{t_1}}, ..., \mathbf{C_{t_N}}) = \frac{1}{N}\sum_{i}^{N} w_i \mathbf{C_{t_i}}$$

(4) Top. The most confident translation:

$$\text{Pseudo}(\mathbf{C_{t_1}}, ..., \mathbf{C_{t_N}}) = \mathbf{C_{t_k}}, k = \arg\max_i w_i$$

Finally, the *BSL* contains a set of English-Chinese word vector pairs, where each entry represents an English social word and its Chinese pseudo-word based on its "translation set".

5.1.5 Constructing the SocVec Space

Let B_i denote the English word of the i^{th} entry of the *BSL*, and its corresponding Chinese pseudo-word is denoted by B_i^*. We can project the English word vector $\mathbf{E_W}$ into the *SocVec* space by computing the cosine similarities between $\mathbf{E_W}$ and each English word vector in *BSL* as values on SocVec dimensions, effectively constructing a new vector $\mathbf{S_W}$ of size L. Similarly, we map a Chinese word vector $\mathbf{C_U}$ to be a new vector $\mathbf{S_U}$. $\mathbf{S_W}$ and $\mathbf{S_U}$ belong to the same vector space *SocVec* and are comparable. The following equation illustrates the projection, and how to compute $ccsim$[1].

$$\begin{aligned} ccsim(W, U) &:= f(\mathbf{E_W}, \mathbf{C_U}) \\ &= sim\left(\begin{bmatrix} cos(\mathbf{E_W}, \mathbf{E_{B_1}}) \\ \vdots \\ cos(\mathbf{E_W}, \mathbf{E_{B_L}}) \end{bmatrix}^{\mathrm{T}}, \begin{bmatrix} cos(\mathbf{C_U}, \mathbf{C_{B_1^*}}) \\ \vdots \\ cos(\mathbf{C_U}, \mathbf{C_{B_L^*}}) \end{bmatrix}^{\mathrm{T}}\right) \\ &= sim(\mathbf{S_W}, \mathbf{S_U}) \end{aligned}$$

For example, if W is "Nagoya" and U is ''名古屋", we compute the cosine similarities between "Nagoya" and each English social word in the *BSL* with their monolingual word embeddings in English. Such similarities compose $\mathbf{S}_{\text{nagoya}}$. Similarly, we compute the cosine similarities between ''名古屋" and each Chinese pseudo-word, and compose the social word vector $\mathbf{S}_{\text{名古屋}}$.

In other words, for each culture/language, the new word vectors like S_W are constructed based on the monolingual similarities of each word to the vectors of a set of task-related words ("social words" in our case). This is also a significant part of the novelty of our transformation method.

5.2 Experimental Setup

Prior to evaluating *SocVec* with our two proposed tasks in Section 5.3 and Section 5.4, we present our preparation steps as follows.

Social Media Corpora Our English Twitter corpus is obtained from Archive Team's Twitter stream grab[2]. The Chinese Weibo corpus comes from Open Weiboscope Data Access[3] [67]. Both corpora cover the whole year of 2012. We then randomly down-sample each corpus to 100 million messages where each message contains at least 10 characters, normalize the text [68], lemmatize the text [69] and use LTP [70] to perform word segmentation for the Chinese corpus.

Entity Linking and Word Embedding Entity linking is a preprocessing step which links various entity mentions (surface forms) to the identity of corresponding entities. For the Twitter corpus, we use Wikifier [71, 72], a widely used entity linker in English. Because no sophisticated tool for Chinese short

[1]The function sim is a generic similarity function, for which several metrics are considered in experiments.

[2]https://archive.org/details/twitterstream

[3]http://weiboscope.jmsc.hku.hk/datazip/

text is available, we implement our own tool that is greedy for high precision. We train English and Chinese monolingual word embedding respectively using *word2vec*'s skip-gram method with a window size of 5 [73].

Bilingual Lexicon Our bilingual lexicon is collected from *Microsoft Translator*[1], which translates English words to multiple Chinese words with confidence scores. Note that all named entities and slang terms used in the following experiments are excluded from this bilingual lexicon.

Social Word Vocabulary Our social word vocabularies come from *Empath* [74] and *OpinionFinder* [75] for English, and *TextMind* [76] for Chinese. Empath is similar to LIWC [64], but has more words and more categories and is publicly available. We manually select 91 categories of words that are relevant to human perception and psychological processes following [66]. OpinionFinder consists of words relevant to opinions and sentiments, and TextMind is a Chinese counterpart for Empath. In summary, we obtain 3,343 words from Empath, 3,861 words from OpinionFinder, and 5,574 unique social words in total.

5.3 Task 1: Mining cross-cultural differences of named entities

Task definition: This task is to discover and quantify cross-cultural differences of concerns towards named entities. Specifically, the input in this task is a list of 700 named entities of interest and two monolingual social media corpora; the output is the scores for the 700 entities indicating the cross-cultural differences of the concerns towards them between two corpora. The ground truth is from the labels collected from human annotators.

5.3.1 Ground Truth Scores

[77] states that the meaning of words is evidenced by the contexts they occur with. Likewise, we assume that the cultural properties of an entity can be captured by the terms they always co-occur within a large social media corpus. Thus, for each of randomly selected 700 named entities, we present human annotators with two lists of 20 most co-occurred terms within Twitter and Weibo corpus respectively.

Our annotators are instructed to rate the topic-relatedness between the two word lists using one of following labels: "very different", "different", "hard to say", "similar" and "very similar". We do this for efficiency and avoiding subjectivity. As the word lists presented come from social media messages, the social and cultural elements are already embedded in their chances of occurrence. All four annotators are native Chinese speakers but have excellent command of English and lived in the US extensively, and they are trained with many selected examples to form shared understanding of the labels. The inter-annotator agreement is 0.67 by Cohen's kappa coefficient, suggesting substantial correlation [36].

5.3.2 Baseline and Our Methods

We propose eight baseline methods for this novel task: **distribution-based** methods (BL-JS, E-BL-JS, and WN-WUP) compute cross-lingual relatedness between two lists of the words surrounding the input English and Chinese terms respectively ($\mathcal{L}_E$ and $\mathcal{L}_C$); **transformation-based** methods (LTrans and BLex) compute

[1] http://www.bing.com/translator/api/Dictionary/Lookup?from=en&to=zh-CHS&text=<input_word>

Table 5–1 Selected culturally different entities with summarized Twitter and Weibo's trending topics

Entity	Twitter topics	Weibo topics
Maldives	coup, president Nasheed quit, political crisis	holiday, travel, honeymoon, paradise, beach
Nagoya	tour, concert, travel, attractive, Osaka	Mayor Takashi Kawamura, Nanjing Massacre, denial of history
Quebec	Conservative Party, Liberal Party, politicians, prime minister, power failure	travel, autumn, maples, study abroad, immigration, independence
Philippines	gunman attack, police, quake, tsunami	South China Sea, sovereignty dispute, confrontation, protest
Yao Ming	NBA, Chinese, good player, Asian	patriotism, collective values, Jeremy Lin, Liu Xiang, Chinese Law maker, gold medal superstar
USC	college football, baseball, Stanford, Alabama, win, lose	top destination for overseas education, Chinese student murdered, scholars, economics, Sino American politics

the vector representation in English and Chinese corpus respectively, and then train a transformation; MCCA, MCluster and Duong are three typical **bilingual word representation models** for computing general cross-lingual word similarities.

The $\mathcal{L}_E$ and $\mathcal{L}_C$ in the BL-JS and WN-WUP methods are the same as the lists that annotators judge. **BL-JS** (*Bilingual Lexicon Jaccard Similarity*) uses the bilingual lexicon to translate $\mathcal{L}_E$ to a Chinese word list $\mathcal{L}_E^*$ as a medium, and then calculates the Jaccard Similarity between $\mathcal{L}_E^*$ and $\mathcal{L}_C$ as J_{EC}. Similarly, we compute J_{CE}. Finally, we regard $(J_{EC}+J_{CE})/2$ as the score of this named entity. **E-BL-JS** (*Embedding-based Jaccard Similarity*) differs from BL-JS in that it instead compares the two lists of words gathered from the rankings of word embedding similarities between the name of entities and all English words and Chinese words respectively. **WN-WUP** (*WordNet Wu-Palmer Similarity*) uses Open Multilingual Wordnet [78] to compute the average similarities over all English-Chinese word pairs constructed from the $\mathcal{L}_E$ and $\mathcal{L}_C$.

We follow the steps of [79] to train a linear transformation (**LTrans**) matrix between *EnVec* and *CnVec*, using 3,000 translation pairs with maximum confidences in the bilingual lexicon. Given a named entity, this solution simply calculates the cosine similarity between the vector of its English name and the *transformed* vector of its Chinese name. **BLex** (*Bilingual Lexicon Space*) is similar to our *SocVec* but it does not use any social word vocabularies but uses bilingual lexicon entries as pivots instead.

MCCA [80] takes two trained monolingual word embeddings with a bilingual lexicon as input, and develop a bilingual word embedding space. It is extended from the work of [81], which performs slightly worse in the experiments. **MCluster** [80] requires re-training the bilingual word embeddings from the two mono-lingual corpora with a bilingual lexicon. Similarly, **Duong** [82] retrains the embeddings from monolingual corpora with an EM-like training algorithm. We also use our BSL as the bilingual lexicon in these methods to investigate its effectiveness and generalizability. The dimensionality is tuned from $\{50, 100, 150, 200\}$ in all these bilingual word embedding methods.

With our constructed ***SocVec*** space, given a named entity with its English and Chinese names, we can simply compute the similarity between their *SocVec*s as its cross-cultural difference score. Our method is

Table 5–2 Comparison of Different Methods

Method	Spearman	Pearson	MAP
BL-JS	0.276	0.265	0.644
WN-WUP	0.335	0.349	0.677
E-BL-JS	0.221	0.210	0.571
LTrans	0.366	0.385	0.644
BLex	0.596	0.595	0.765
MCCA-BL(100d)	0.325	0.343	0.651
MCCA-BSL(150d)	0.357	0.376	0.671
MCluster-BL(100d)	0.365	0.388	0.693
MCluster-BSL(100d)	0.391	0.425	0.713
Duong-BL(100d)	0.618	0.627	0.785
Duong-BSL(100d)	0.625	0.631	0.791
SocVec:opn	0.668	0.662	**0.834**
SocVec:all	**0.676**	**0.671**	**0.834**
SocVec:noun	0.564	0.562	0.756
SocVec:verb	0.615	0.618	0.779
SocVec:adj.	0.636	0.639	0.800

based on monolingual word embeddings and a BSL, and thus does not need the time-consuming re-training on the corpora.

5.3.3 Experimental Results

For qualitative evaluation, Table 5–1 shows some of the most culturally different entities mined by the SocVec method. The hot and trendy topics on Twitter and Weibo are manually summarized to help explain the cross-cultural differences. The perception of these entities diverges widely between English and Chinese social media, thus suggesting significant cross-cultural differences. Note that some cultural differences are time-specific. We believe such temporal variations of cultural differences can be valuable and beneficial for social studies as well. Investigating temporal factors of cross-cultural differences in social media can be an interesting future research topic in this task.

In Table 5–2, we evaluate the benchmark methods and our approach with three metrics: Spearman and Pearson, where correlation is computed between truth averaged scores (quantifying the labels from 1.0 to 5.0) and computed cultural difference scores from different methods; Mean Average Precision (MAP), which converts averaged scores as binary labels, by setting 3.0 as the threshold. The ***SocVec:opn*** considers only OpinionFinder as the ESV, while ***SocVec:all*** uses the union of Empath and OpinionFinder vocabularies[1].

Lexicon Ablation Test. To show the effectiveness of social words versus other type of words as the bridge between the two cultures, we also compare the results using sets of nouns (***SocVec:noun***), verbs (***SocVec:verb***) and adjectives (***SocVec:adj.***). All vocabularies under comparison are of similar sizes (around

[1]The following tuned parameters are used in *SocVec* methods: 5-word context window, 150 dimensions monolingual word vectors, cosine similarity as the *sim* function, and "*Top*" as the pseudo-word generator.

Table 5–3 Different Similarity Functions

Similarity	Spearman	Pearson	MAP
PCorr.	0.631	0.625	0.806
L1 + M	0.666	0.656	0.824
Cos	**0.676**	0.669	**0.834**
L2 + E	**0.676**	**0.671**	**0.834**

Table 5–4 Different Pseudo-word Generators

Generator	Spearman	Pearson	MAP
Max.	0.413	0.401	0.726
Avg.	0.667	0.625	0.831
W.Avg.	0.671	0.660	0.832
Top	**0.676**	**0.671**	**0.834**

5,000), indicating that the improvement of our method is significant. Results show that our *SocVec* models, and in particular, the *SocVec* model using the social words as cross-lingual media, performs the best.

Similarity Options. We also evaluate the effectiveness of four different similarity options in *SocVec*, namely, Pearson Correlation Coefficient (*PCorr.*), L1-normalized Manhattan distance (*L1+M*), Cosine Similarity (*Cos*) and L2-normalized Euclidean distance (*L2+E*). From Table 5–3, we conclude that among these four options, *Cos* and *L2+E* perform the best.

Pseudo-word Generators. Table 5–4 shows effect of using four pseudo-word generator functions, from which we can infer that "*Top*" generator function performs best for it reduces some noisy translation pairs.

5.4 Task 2: Finding most similar words for slang across languages

Task Description: This task is to find the most similar English words of a given Chinese slang term in terms of its slang meanings and sentiment, and vice versa. The input is a list of English/Chinese slang terms of interest and two monolingual social media corpora; the output is a list of Chinese/English word sets corresponding to each input slang term. Simply put, for each given slang term, we want to find a set of the words in a different language that are most similar to itself and thus can help people understand it across languages. We propose Average Cosine Similarity (Section 5.4.3) to evaluate a method's performance with the ground truth (presented below).

5.4.1 Ground Truth

Slang Terms. We collect the Chinese slang terms from an online Chinese slang glossary[1] consisting of 200 popular slang terms with English explanations. For English, we resort to a slang word list from OnlineSlangDictionary[2] with explanations and downsample the list to 200 terms.

[1] https://www.chinasmack.com/glossary
[2] http://onlineslangdictionary.com/word-list/

Table 5–5 ACS Sum Results of Slang Translation

Gg	Bi	Bd	CC	LT
18.24	16.38	17.11	17.38	9.14
TransBL	MCCA	MCluster	Duong	SV
18.13	17.29	17.47	20.92	**23.01**

(a) Chinese Slang to English

Gg	Bi	Bd	LT	TransBL
6.40	15.96	15.44	7.32	11.43
MCCA	MCluster	Duong	SV	
15.29	14.97	15.13	**17.31**	

(b) English Slang to Chinese

Truth Sets. For each Chinese slang term, its truth set is a set of words extracted from its English explanation. For example, we construct the truth set of the Chinese slang term ‘‘二百五” by manually extracting significant words about its slang meanings (bold) in the glossary:

二百五: A ***foolish*** person who is lacking in sense but still ***stubborn***, ***rude***, and ***impetuous***.

Similarly, for each English slang term, its Chinese word sets are the translation of the words hand picked from its English explanation.

Table 5–6 Bidirectional Slang Translation Examples Produced by SocVec

Slang	**Explanation**	**Google**	**Bing**	**Baidu**	**Ours**
浮云	something as ephemeral and unimportant as “passing clouds”	clouds	nothing	floating clouds	nothingness, illusion
水军	“water army”, people paid to slander competitors on the Internet and to help shape public opinion	Water army	Navy	Navy	propaganda, complicit, fraudulent
floozy	a woman with a reputation for promiscuity	N/A	劣根性 (depravity)	荡妇 (slut)	骚货 (slut), 妖精 (promiscuous)
fruitcake	a crazy person, someone who is completely insane	水果蛋糕 (fruit cake)	水果蛋糕 (fruit cake)	水果蛋糕 (fruit cake)	怪诞 (bizarre), 厌烦 (annoying)

5.4.2 Baseline and Our Methods

We propose two types of baseline methods for this task. The first is based on well-known *online translators*, namely Google (Gg), Bing (Bi) and Baidu (Bd). Note that experiments using them are done in August, 2017. Another baseline method for Chinese is CC-CEDICT[1] (CC), an online public Chinese-English dictionary, which is constantly updated for popular slang terms.

Considering situations where many slang terms have literal meanings, it may be unfair to retrieve target terms from such machine translators by solely inputing slang terms without specific contexts. Thus, we

[1] https://cc-cedict.org/wiki/

utilize example sentences of their slang meanings from some websites (mainly from Urban Dictionary[1]). The following example shows how we obtain the target translation terms for the slang word "fruitcake" (an insane person):

Input sentence: *Oh man, you don't want to date that girl. She's always drunk and yelling. She is a total* ***fruitcake***.[2]

Google Translation: 哦, 男人, 你不想约会那个女孩。她总是喝醉了, 大喊大叫。她是一个**水果蛋糕**。

Another lines of baseline methods is scoring-based. The basic idea is to score all words in our bilingual lexicon and consider the top K words as the target terms. Given a source term to be translated, the Linear Transform (LT), MCCA, MCluster and Duong methods score the candidate target terms by computing cosine similarities in their constructed bilingual vector space (with the tuned best settings in previous evaluation). A more sophisticated baseline (TransBL) leverages the bilingual lexicon: for each candidate target term w in the target language, we first obtain its translations T_w back into the source language and then calculate the average word similarities between the source term and the translations T_w as w's score.

Our *SocVec-based method* (**SV**) is also scoring-based. It simply calculates the cosine similarities between the source term and each candidate target term within *SocVec* space as their scores.

5.4.3 Experimental Results

To quantitatively evaluate our methods, we need to measure similarities between a produced word set and the ground truth set. Exact-matching Jaccard similarity is too strict to capture valuable relatedness between two word sets. We argue that average cosine similarity (ACS) between two sets of word vectors is a better metric for evaluating the similarity between two word sets.

$$\mathrm{ACS}(A,B) = \frac{1}{|A||B|}\sum_{i=1}^{|A|}\sum_{j=1}^{|B|}\frac{\mathbf{A_i}\cdot\mathbf{B_j}}{\|\mathbf{A_i}\|\|\mathbf{B_j}\|}$$

The above equation illustrates such computation, where A and B are the two word sets: A is the truth set and B is a similar list produced by each method. In the previous case of ''二百五" (Section 5.4.1), A is {foolish, stubborn, rude, impetuous} while B can be {imbecile, brainless, scumbag, imposter}. $\mathbf{A_i}$ and $\mathbf{B_j}$ denote the word vector of the i^{th} word in A and j^{th} word in B respectively. The embeddings used in ACS computations are pre-trained *GloVe* word vectors[3] and thus the computation is fair among different methods.

Experimental results of Chinese and English slang translation in terms of the sum of *ACS* over 200 terms are shown in Table 5–5. The performance of online translators for slang typically depends on human-set rules and supervised learning on well-annotated parallel corpora, which are rare and costly, especially for social media where slang emerges the most. This is probably the reason why they do not perform well. The Linear Transformation (LT) model is trained on highly confident translation pairs in the bilingual lexicon, which lacks OOV slang terms and social contexts around them. The TransBL method is competitive because its similarity computations are within monolingual semantic spaces and it makes great use of the bilingual

[1] http://www.urbandictionary.com/
[2] http://www.englishbaby.com/lessons/4349/slang/fruitcake
[3] https://nlp.stanford.edu/projects/glove/

Table 5–7 Slang-to-Slang Translation Examples

Chinese Slang	English Slang	Explanation
萌	adorbz, adorb, adorbs, tweeny, attractiveee	cute, adorable
二百五	shithead, stupidit, douchbag	A foolish person
鸭梨	antsy, stressy, fidgety, grouchy, badmood	stress, pressure, burden

lexicon, but it loses the information from the related words that are not in the bilingual lexicon. Our method (SV) outperforms baselines by directly using the distances in the *SocVec* space, which proves that the *SocVec* well captures the cross-cultural similarities between terms.

To qualitatively evaluate our model, in Table 5–6, we present several examples of our translations for Chinese and English slang terms as well as their explanations from the glossary. Our results are highly correlated with these explanations and capture their significant semantics, whereas most online translators just offer literal translations, even within obviously slang contexts. We take a step further to directly translate Chinese slang terms to English slang terms by filtering out ordinary (non-slang) words in the original target term lists, with examples shown in Table 5–7.

5.5 Related Work

Although social media messages have been essential resources for research in computational social science, most works based on them only focus on a single culture and language [83–88]. Cross-cultural studies have been conducted on the basis of a questionnaire-based approach for many years. There are only a few of such studies using NLP techniques.

[89] present a framework to visualize the cross-cultural differences in concerns in multilingual blogs collected with a topic keyword. [65] show that cross-cultural analysis through language in social media data is effective, especially using emotion terms as culture features, but the work is restricted in monolingual analysis and a single domain (love and relationship). [66] investigate the cross-cultural differences in word usages between Australian and American English through their proposed "socio-linguistic features" (similar to our social words) in a supervised way. With the data of social network structures and user interactions, [90] study how to quantify the controversy of topics within a culture and language. [91] propose an approach to detect differences of word usage in the cross-lingual topics of multilingual topic modeling results. To the best of our knowledge, our work for Task 1 is among the first to mine and quantify the cross-cultural differences in concerns about named entities across different languages.

Existing research on slang mainly focuses on automatic discovering of slang terms [92] and normalization of noisy texts [68] as well as slang formation.

parenciteni2017learning are among the first to propose an automatic supervised framework to mono-lingually explain slang terms using external resources. However, research on automatic translation or cross-lingually explanation for slang terms is missing from the literature. Our work in Task 2 fills the gap by computing cross-cultural similarities with our bilingual word representations (*SocVec*) in an unsupervised way. We believe this application is useful in machine translation for social media [93].

Many existing cross-lingual word embedding models rely on expensive parallel corpora with word or sentence alignments [28, 94]. These works often aim to improve the performance on monolingual tasks and cross-lingual model transfer for document classification, which does not require cross-cultural signals. We position our work in a broader context of "monolingual mapping" based cross-lingual word embedding models in the survey of [25]. The SocVec uses only lexicon resource and maps monolingual vector spaces into a common high-dimensional third space by incorporating social words as pivot, where orthogonality is approximated by setting clear meaning to each dimension of the *SocVec* space.

5.6 Conclusion

We present the SocVec method to compute cross-cultural differences and similarities, and evaluate it on two novel tasks about mining cross-cultural differences in named entities and computing cross-cultural similarities in slang terms. Through extensive experiments, we demonstrate that the proposed lightweight yet effective method outperforms a number of baselines, and can be useful in translation applications and cross-cultural studies in computational social science. Future directions include: 1) mining cross-cultural differences in general concepts other than names and slang, 2) merging the mined knowledge into existing knowledge bases, and 3) applying the SocVec in downstream tasks like machine translation.

Summary

Commonsense knowledge and related works have been one of the most important areas in Artificial Intelligence, because a lot of artificial intelligent systems can benefit from incorporating commonsense knowledge as background priors in their models. These kinds of commonsense facts have been used in many downstream tasks, such as textual entailment in Natural Language Processing (NLP) and object detection in Computer Vision (CV).

An automatic method of extracting commonsense relationship from textual corpora or other data is an essential topic. LocatedNear relation is a kind of commonsense knowledge describing two physical objects that are typically found near each other in real life, of which ConceptNet contains only 49 triples. In the first section of this thesis, the author studies how to automatically extract such relationship through a sentence-level relation classifier and aggregating the scores of entity pairs from a large corpus. Apart from that, we release two benchmark datasets for evaluation and future research: 1) one containing 5,000 sentences annotated with whether a mentioned entity pair has LocatedNear relation in the given sentence or not; 2) the other containing 500 pairs of physical objects and whether they are commonly located nearby. We propose a number of baseline methods for the tasks and compare the results with a state-of-the-art general-purpose relation classifier. The second section of this thesis proposes the very first dataset for CSKGE and investigate the characteristics as well as the performance of state-of-the-art KGE models on it. The author also proposes a novel CSKGE model purposely designed for CSKGs. The third section of this thesis studies the problem of computing such cross-cultural differences and similarities. This thesis presents a lightweight yet effective approach, and evaluate it on two novel tasks: 1) mining cross-cultural differences of named entities and 2) finding similar terms for slang across languages. Experimental results show that our framework substantially outperforms a number of baseline methods on both tasks. The framework could be useful for machine translation applications and research in computational social science.

The works in this thesis are published in the following papers:

1. Mining Cross-Cultural Differences and Similarities in Social Media.
   (to appear) in Proc. of *ACL 2018* (CCF-A class)
   Bill Yuchen Lin, Frank F. Xu, Kenny Q. Zhu, Seung-won Hwang
2. Automatic Extraction of Commonsense LocatedNear Knowledge.
   (to appear) in Proc. of *ACL 2018* (CCF-A class)
   Frank F. Xu, **Bill Yuchen Lin**, Kenny Q. Zhu
3. Multi-channel BiLSTM-CRF Model for Emerging Named Entity Recognition in Social Media.
   in Proc. of *EMNLP 2017 (CCF-B class) Workshop on Noisy User-generated Text*
   Bill Y. Lin, Frank F. Xu, Zhiyi Luo, Kenny Q. Zhu

Bibliography

[1] DAGAN I, DOLAN B, MAGNINI B, et al. Recognizing textual entailment: Rational, evaluation and approaches[J]. Natural Language Engineering, 2009, 15(4): i–xvii. DOI: 10.1017/S1351324909990234.

[2] BOWMAN S R, ANGELI G, POTTS C, et al. A large annotated corpus for learning natural language inference[C/OL]// Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Lisbon, Portugal: Association for Computational Linguistics, 2015: 632–642. http://www.aclweb.org/anthology/D15-1075. DOI: 10.18653/v1/D15-1075.

[3] ZHU Y, FATHI A, FEI-FEI L. Reasoning about object affordances in a knowledge base representation[C]// European conference on computer vision. Springer. [S.l.]: [s.n.], 2014: 408–424. DOI: 10.1007/978-3-319-10605-2_27.

[4] SPEER R, HAVASI C. Representing General Relational Knowledge in ConceptNet 5[C/OL]// Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012). Istanbul, Turkey: European Language Resources Association (ELRA), 2012. http://www.aclweb.org/anthology/L12-1639.

[5] YATSKAR M, ORDONEZ V, FARHADI A. Stating the Obvious: Extracting Visual Common Sense Knowledge[C/OL]// Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. San Diego, California: Association for Computational Linguistics, 2016: 193–198. http://www.aclweb.org/anthology/N16-1023. DOI: 10.18653/v1/N16-1023.

[6] LI X, TAHERI A, TU L, et al. Commonsense knowledge base completion[C/OL]// Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL), Berlin, Germany, August. Association for Computational Linguistics. [S.l.]: [s.n.], 2016: 1445–1455. http://www.aclweb.org/anthology/P16-1137. DOI: 10.18653/v1/P16-1137.

[7] BOLLACKER K D, EVANS C, PARITOSH P, et al. Freebase: a collaboratively created graph database for structuring human knowledge[C]// SIGMOD Conference. [S.l.]: [s.n.], 2008.

[8] MILLER G A. WORDNET: A Lexical Database for English[J]. Commun. ACM, 1992, 38: 39–41.

[9] BORDES A, USUNIER N, GARCÍA-DURÁN A, et al. Translating Embeddings for Modeling Multi-relational Data[C]// NIPS. [S.l.]: [s.n.], 2013.

[10] WANG Z, ZHANG J, FENG J, et al. Knowledge Graph Embedding by Translating on Hyperplanes[C]// AAAI. [S.l.]: [s.n.], 2014.

[11] LIN Y, LIU Z, SUN M, et al. Learning Entity and Relation Embeddings for Knowledge Graph Completion[C]// AAAI. [S.l.]: [s.n.], 2015.

[12] YANG B, YIH W.-T, HE X, et al. Embedding Entities and Relations for Learning and Inference in Knowledge Bases[J]. CoRR, 2014, abs/1412.6575.

[13] TROUILLON T, WELBL J, RIEDEL S, et al. Complex Embeddings for Simple Link Prediction[C]//ICML. [S.l.]: [s.n.], 2016.

[14] ZHONG H, ZHANG J, WANG Z, et al. Aligning Knowledge and Text Embeddings by Entity Descriptions[C]//EMNLP. [S.l.]: [s.n.], 2015.

[15] XIE R, LIU Z, JIA J, et al. Representation Learning of Knowledge Graphs with Entity Descriptions[C]//AAAI. [S.l.]: [s.n.], 2016.

[16] XU J, QIU X, CHEN K, et al. Knowledge Graph Representation with Jointly Structural and Textual Encoding[C]//IJCAI. [S.l.]: [s.n.], 2017.

[17] SPEER R, HAVASI C. Representing General Relational Knowledge in ConceptNet 5[C]//LREC. [S.l.]: [s.n.], 2012.

[18] CARO L D, RUGGERI A, CUPI L, et al. Common-Sense Knowledge for Natural Language Understanding: Experiments in Unsupervised and Supervised Settings[C]//AI*IA. [S.l.]: [s.n.], 2015.

[19] AGARWAL B, MITTAL N, BANSAL P, et al. Sentiment Analysis Using Common-Sense and Context Information[C]//Comp. Int. and Neurosc. [S.l.]: [s.n.], 2015.

[20] WANG Z, ZHANG J, FENG J, et al. Knowledge Graph and Text Jointly Embedding[C]//EMNLP. [S.l.]: [s.n.], 2014.

[21] SOCHER R, CHEN D, MANNING C D, et al. Reasoning With Neural Tensor Networks for Knowledge Base Completion[C]//NIPS. [S.l.]: [s.n.], 2013.

[22] BORDES A, GLOROT X, WESTON J, et al. A semantic matching energy function for learning with multi-relational data[J]. Machine Learning, 2013, 94: 233–259.

[23] PENNINGTON J, SOCHER R, MANNING C. Glove: Global Vectors for Word Representation[C/OL]//Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Doha, Qatar: Association for Computational Linguistics, 2014: 1532–1543. http://www.aclweb.org/anthology/D14-1162. DOI: 10.3115/v1/D14-1162.

[24] LE Q V, MIKOLOV T. Distributed Representations of Sentences and Documents[C]//Proc. of ICML. [S.l.]: [s.n.], 2014. DOI: 10.1.1.646.3937.

[25] RUDER S, VULI I, SØGAARD A. A survey of cross-lingual embedding models[J/OL]. ArXiv preprint arXiv:1706.04902, 2017. https://arxiv.org/pdf/1706.04902.pdf.

[26] CAMACHO-COLLADOS J, PILEHVAR M T, COLLIER N, et al. SemEval-2017 Task 2: Multilingual and Cross-lingual Semantic Word Similarity[C/OL]//Proc. of SemEval@ACL. [S.l.]: [s.n.], 2017. http://www.aclweb.org/anthology/S17-2002. DOI: 10.18653/v1/S17-2002.

[27] SARATH C A P, LAULY S, LAROCHELLE H, et al. An Autoencoder Approach to Learning Bilingual Word Representations[C/OL]// Proc. in NIPS. [S.l.]: [s.n.], 2014. https://papers.nips.cc/paper/5270-an-autoencoder-approach-to-learning-bilingual-word-representations.pdf.

[28] KOISKÝ T, HERMANN K M, BLUNSOM P. Learning Bilingual Word Representations by Marginalizing Alignments[C/OL]// Proc. of ACL. [S.l.]: [s.n.], 2014. http://www.aclweb.org/anthology/P14-2037. DOI: 10.3115/v1/P14-2037.

[29] UPADHYAY S, FARUQUI M, DYER C, et al. Cross-lingual models of word embeddings: An empirical comparison[C/OL]// Proc. of ACL. [S.l.]: [s.n.], 2016. http://www.aclweb.org/anthology/P16-1157. DOI: 10.18653/v1/P16-1157.

[30] XU Y, MOU L, LI G, et al. Classifying Relations via Long Short Term Memory Networks along Shortest Dependency Paths[C/OL]// Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Lisbon, Portugal: Association for Computational Linguistics, 2015: 1785–1794. http://www.aclweb.org/anthology/D15-1206. DOI: 10.18653/v1/D15-1206.

[31] ZENG D, LIU K, LAI S, et al. Relation Classification via Convolutional Deep Neural Network[C/OL]// Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers. Dublin, Ireland: Dublin City University, Association for Computational Linguistics, 2014: 2335–2344. http://www.aclweb.org/anthology/C14-1220.

[32] ZAREMBA W, SUTSKEVER I, VINYALS O. Recurrent neural network regularization[J/OL]. ArXiv preprint arXiv:1409.2329, 2014. https://arxiv.org/abs/1409.2329.

[33] LAHIRI S. Complexity of Word Collocation Networks: A Preliminary Structural Analysis[C/OL]// Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics. Gothenburg, Sweden: Association for Computational Linguistics, 2014: 96–105. http://www.aclweb.org/anthology/E14-3011. DOI: 10.3115/v1/E14-3011.

[34] MINTZ M, BILLS S, SNOW R, et al. Distant supervision for relation extraction without labeled data[C/OL]// 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. Suntec, Singapore: Association for Computational Linguistics, 2009: 1003–1011. http://www.aclweb.org/anthology/P09-1113.

[35] RIEDEL S, YAO L, MCCALLUM A. Modeling relations and their mentions without labeled text[J/OL]. Machine learning and knowledge discovery in databases, 2010: 148–163. https://link.springer.com/content/pdf/10.1007/978-3-642-15939-8_10.pdf.

[36] LANDIS J R, KOCH G G. The measurement of observer agreement for categorical data.[J]. Biometrics, 1977, 33 1: 159–74. DOI: 10.2307/2529310.

[37] HENDRICKX I, KIM S N, KOZAREVA Z, et al. SemEval-2010 Task 8: Multi-Way Classification of Semantic Relations between Pairs of Nominals[C/OL]// Proceedings of the 5th International Workshop on Semantic Evaluation. Uppsala, Sweden: Association for Computational Linguistics, 2010: 33–38. http://www.aclweb.org/anthology/S10-1006.

[38] XU Y, JIA R, MOU L, et al. Improved relation classification by deep recurrent neural networks with data augmentation[C/OL]// Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers. Osaka, Japan: The COLING 2016 Organizing Committee, 2016: 1461–1470. http://www.aclweb.org/anthology/C16-1138.

[39] SOCHER R, PENNINGTON J, HUANG E H, et al. Semi-supervised recursive autoencoders for predicting sentiment distributions[C]// Proceedings of the conference on empirical methods in natural language processing. Association for Computational Linguistics. [S.l.]: [s.n.], 2011: 151–161.

[40] EBRAHIMI J, DOU D. Chain Based RNN for Relation Classification.[C]// HLT-NAACL. [S.l.]: [s.n.], 2015: 1244–1249.

[41] LIN T.-Y, MAIRE M, BELONGIE S, et al. Microsoft coco: Common objects in context[C]// European Conference on Computer Vision. Springer. [S.l.]: [s.n.], 2014: 740–755.

[42] SPEER R, CHIN J, HAVASI C. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge[C]// AAAI. [S.l.]: [s.n.], 2017.

[43] DUCHI J C, HAZAN E, SINGER Y. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization[J]. Journal of Machine Learning Research, 2010, 12: 2121–2159.

[44] PENNINGTON J, SOCHER R, MANNING C D. Glove: Global Vectors for Word Representation[C]// EMNLP. [S.l.]: [s.n.], 2014.

[45] WANG Z, LI J.-Z. Text-Enhanced Representation Learning for Knowledge Graph[C]// IJCAI. [S.l.]: [s.n.], 2016.

[46] XIAO H, HUANG M, MENG L, et al. SSP: Semantic Space Projection for Knowledge Graph Embedding with Text Descriptions[C]// AAAI. [S.l.]: [s.n.], 2017.

[47] HAN X, LIU Z, SUN M. Neural Knowledge Acquisition via Mutual Attention between Knowledge Graph and Text[J]. 2018.

[48] BALDWIN T, KIM Y.-B, DE MARNEFFE M C, et al. Shared tasks of the 2015 workshop on noisy user-generated text: Twitter lexical normalization and named entity recognition[J]. ACL-IJCNLP, 2015, 126: 2015.

[49] DERCZYNSKI L, MAYNARD D, RIZZO G, et al. Analysis of named entity recognition and linking for tweets[J]. Information Processing & Management, 2015, 51(2): 32–49.

[50] AUGENSTEIN I, DERCZYNSKI L, BONTCHEVA K. Generalisation in named entity recognition: A quantitative analysis[J]. Computer Speech & Language, 2017, 44: 61–83.

[51] ZHANG B, HUANG H, PAN X, et al. Context-aware Entity Morph Decoding[C]//Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers. [S.l.]: [s.n.], 2015: 586–595.

[52] MA X, HOVY E H. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF[C]//Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers. [S.l.]: [s.n.], 2016.

[53] LAMPLE G, BALLESTEROS M, SUBRAMANIAN S, et al. Neural Architectures for Named Entity Recognition[C]//NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016. [S.l.]: [s.n.], 2016: 260–270.

[54] OWOPUTI O, O'CONNOR B, DYER C, et al. Improved part-of-speech tagging for online conversational text with word clusters[C]//. Association for Computational Linguistics. [S.l.]: [s.n.], 2013.

[55] KONG L, SCHNEIDER N, SWAYAMDIPTA S, et al. A dependency parser for tweets[J]. 2014.

[56] GIMPEL K, SCHNEIDER N, O'CONNOR B, et al. Part-of-speech tagging for twitter: Annotation, features, and experiments[C]//Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2. Association for Computational Linguistics. [S.l.]: [s.n.], 2011: 42–47.

[57] XU Y, JIA R, MOU L, et al. Improved relation classification by deep recurrent neural networks with data augmentation[J]. ArXiv preprint arXiv:1601.03651, 2016.

[58] HE K, ZHANG X, REN S, et al. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification[C]//2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015. [S.l.]: [s.n.], 2015: 1026–1034.

[59] GLOROT X, BENGIO Y. Understanding the difficulty of training deep feedforward neural networks[C]//Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2010, Chia Laguna Resort, Sardinia, Italy, May 13-15, 2010. [S.l.]: [s.n.], 2010: 249–256.

[60] JÓZEFOWICZ R, ZAREMBA W, SUTSKEVER I. An Empirical Exploration of Recurrent Network Architectures[C]//Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015. [S.l.]: [s.n.], 2015: 2342–2350.

[61] KINGMA D, BA J. Adam: A method for stochastic optimization[J]. ArXiv preprint arXiv:1412.6980, 2014.

[62] KITAYAMA S, MARKUS H R, KUROKAWA M. Culture, emotion, and well-being: Good feelings in Japan and the United States[J]. Cognition & Emotion, 2000, 14(1): 93–124. DOI: 10.1080/026999300379003.

[63] GAREIS E, WILKINS R. Love expression in the United States and Germany[J/OL]. International Journal of Intercultural Relations, 2011, 35(3): 307–319. https://doi.org/10.1016%2Fj.ijintrel.2010.06.006. DOI: 10.1016/j.ijintrel.2010.06.006.

[64] TAUSCZIK Y R, PENNEBAKER J W. The Psychological Meaning of Words: LIWC and Computerized Text Analysis Methods[J/OL]. Journal of Language and Social Psychology, 2009, 29(1): 24–54. https://doi.org/10.1177%2F0261927x09351676. DOI: 10.1177/0261927x09351676.

[65] ELAHI M F, MONACHESI P. An Examination of Cross-Cultural Similarities and Differences from Social Media Data with respect to Language Use.[C/OL]// Proc. of LREC. [S.l.]: [s.n.], 2012. http://www.lrec-conf.org/proceedings/lrec2012/pdf/942_Paper.pdf.

[66] GARIMELLA A, MIHALCEA R, PENNEBAKER J W. Identifying Cross-Cultural Differences in Word Usage[C/OL]// Proc. of COLING. [S.l.]: [s.n.], 2016. http://www.aclweb.org/anthology/C16-1065.

[67] FU K.-W, CHAN C.-H, CHAU M. Assessing censorship on microblogs in China: Discriminatory keyword analysis and the real-name registration policy[J]. IEEE Internet Computing, 2013, 17(3): 42–50. DOI: 10.1109/MIC.2013.28.

[68] HAN B, COOK P, BALDWIN T. Automatically constructing a normalisation dictionary for microblogs[C/OL]// Proc. of EMNLP-CoNLL. [S.l.]: [s.n.], 2012. http://www.aclweb.org/anthology/D12-1039.

[69] MANNING C D, SURDEANU M, BAUER J, et al. The stanford corenlp natural language processing toolkit.[C/OL]// Proc. of ACL (System Demonstrations). [S.l.]: [s.n.], 2014. http://www.aclweb.org/anthology/P14-5010. DOI: 10.3115/v1/P14-5010.

[70] CHE W, LI Z, LIU T. Ltp: A chinese language technology platform[C/OL]// Proc. of COLING 2010: Demonstrations. [S.l.]: [s.n.], 2010. http://www.aclweb.org/anthology/C10-3004.

[71] RATINOV L, ROTH D, DOWNEY D, et al. Local and global algorithms for disambiguation to wikipedia[C/OL]// Proc. of ACL. [S.l.]: [s.n.], 2011. http://www.aclweb.org/anthology/P11-1138.

[72] CHENG X, ROTH D. Relational Inference for Wikification[C/OL]// Proc. of EMNLP. [S.l.]: [s.n.], 2013. http://www.aclweb.org/anthology/D13-1184.

[73] MIKOLOV T, SUTSKEVER I, CHEN K, et al. Distributed representations of words and phrases and their compositionality[C/OL]// Proc. of NIPS. [S.l.]: [s.n.], 2013. http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf.

[74] FAST E, CHEN B, BERNSTEIN M S. Empath: Understanding topic signals in large-scale text[C]// Proc. of CHI. [S.l.]: [s.n.], 2016. DOI: 10.1145/2858036.2858535.

[75] CHOI Y, CARDIE C, RILOFF E, et al. Identifying sources of opinions with conditional random fields and extraction patterns[C/OL]// Proc. of HLT-EMNLP. [S.l.]: [s.n.], 2005. http://www.aclweb.org/anthology/H05-1045.

[76] GAO R, HAO B, LI H, et al. Developing simplified Chinese psychological linguistic analysis dictionary for microblog[C]// Proceedings of International Conference on Brain and Health Informatics. Springer. [S.l.]: [s.n.], 2013. DOI: 10.1007/978-3-319-02753-1_36.

[77] HARRIS Z S. Distributional structure[J]. Word, 1954, 10(2-3): 146–162. DOI: 10.1080/00437956.1954.11659520.

[78] WANG S, BOND F. Building the chinese open wordnet (cow): Starting from core synsets[C/OL]// Proceedings of the 11th Workshop on Asian Language Resources, a Workshop at IJCNLP. [S.l.]: [s.n.], 2013. http://www.aclweb.org/anthology/W13-4302.

[79] MIKOLOV T, LE Q V, SUTSKEVER I. Exploiting similarities among languages for machine translation[J]. ArXiv preprint arXiv:1309.4168, 2013. DOI: 10.1.1.754.2995.

[80] AMMAR W, MULCAIRE G, TSVETKOV Y, et al. Massively multilingual word embeddings[J/OL]. ArXiv preprint arXiv:1602.01925, 2016. https://arxiv.org/pdf/1602.01925.pdf.

[81] FARUQUI M, DYER C. Improving Vector Space Word Representations Using Multilingual Correlation[C/OL]// Proc. of EACL. [S.l.]: [s.n.], 2014. http://aclweb.org/anthology/E/E14/E14-1049.pdf.

[82] DUONG L, KANAYAMA H, MA T, et al. Learning Crosslingual Word Embeddings without Bilingual Corpora[C/OL]// Proc. of EMNLP. [S.l.]: [s.n.], 2016. http://www.aclweb.org/anthology/D16-1136. DOI: 10.18653/v1/D16-1136.

[83] PETROVIC S, OSBORNE M, LAVRENKO V. Streaming First Story Detection with application to Twitter[C/OL]// Proc. of HLT-NAACL. [S.l.]: [s.n.], 2010. http://www.aclweb.org/anthology/N10-1021.

[84] PAUL M J, DREDZE M. You Are What You Tweet: Analyzing Twitter for Public Health[C/OL]// Proc. of ICWSM. [S.l.]: [s.n.], 2011. http://www.cs.jhu.edu/~mpaul/files/2011.icwsm.twitter_health.pdf.

[85] ROSENTHAL S, MCKEOWN K. I Couldn't Agree More: The Role of Conversational Structure in Agreement and Disagreement Detection in Online Discussions[C]// Proc. of SIGDIAL. [S.l.]: [s.n.], 2015. DOI: 10.18653/v1/W15-4625.

[86] WANG W Y, YANG D. That's So Annoying!!!: A Lexical and Frame-Semantic Embedding Based Data Augmentation Approach to Automatic Categorization of Annoying Behaviors using #petpeeve Tweets[C/OL]// Proc. of EMNLP. [S.l.]: [s.n.], 2015. http://www.aclweb.org/anthology/D15-1306. DOI: 10.18653/v1/D15-1306.

[87] ZHANG B, HUANG H, PAN X, et al. Context-aware Entity Morph Decoding[C/OL]// Proc. of ACL. [S.l.]: [s.n.], 2015. http://www.aclweb.org/anthology/P15-1057. DOI: 10.3115/v1/P15-1057.

[88] LIN B Y, XU F F, LUO Z, et al. Multi-channel BiLSTM-CRF Model for Emerging Named Entity Recognition in Social Media[C/OL]// Proc. of W-NUT@EMNLP. [S.l.]: [s.n.], 2017. https://aclanthology.info/papers/W17-4421/w17-4421. DOI: 10.18653/v1/w17-4421.

[89] NAKASAKI H, KAWABA M, YAMAZAKI S, et al. Visualizing Cross-Lingual/Cross-Cultural Differences in Concerns in Multilingual Blogs.[C/OL]// Proc. of ICWSM. [S.l.]: [s.n.], 2009. https://pdfs.semanticscholar.org/484f/9d44345015338e49c59d0a67210c276f7707.pdf?_ga=2.209852949.1831208069.1525515737-2139274556.1519386756.

[90] GARIMELLA K, MORALES G D F, GIONIS A, et al. Quantifying Controversy in Social Media[C]// Proc. of WSDM. [S.l.]: [s.n.], 2016. DOI: 10.1145/2835776.2835792.

[91] GUTIÉRREZ E D, SHUTOVA E, LICHTENSTEIN P, et al. Detecting Cross-cultural Differences Using a Multilingual Topic Model[J/OL]. TACL, 2016, 4: 47–60. http://www.aclweb.org/anthology/Q16-1004.

[92] ELSAHAR H, ELBELTAGY S R. A Fully Automated Approach for Arabic Slang Lexicon Extraction from Microblogs[C]// Proc. of CICLing. [S.l.]: [s.n.], 2014. DOI: 10.1007/978-3-642-54906-9_7.

[93] LING W, XIANG G, DYER C, et al. Microblogs as Parallel Corpora[C/OL]// Proc. of ACL. [S.l.]: [s.n.], 2013. http://www.aclweb.org/anthology/P13-1018.

[94] KLEMENTIEV A, TITOV I, BHATTARAI B. Inducing Crosslingual Distributed Representations of Words[C/OL]// Proc. of COLING. [S.l.]: [s.n.], 2012. http://www.aclweb.org/anthology/C12-1089.

Acknowledgements

First and foremost, I want to express my profound gratitude to my supervisor Prof. Kenny Zhu for his support of my undergraduate research in these two years, for his immense knowledge and continuous encouragement. His mentoring helped me in all the time of research and writing of my publications. He showed me what good research should be and taught me how to do it, which was the most important thing I have ever learned in my undergraduate study.

My sincere thanks also goes to Frank Xu, my best friend and best co-author, who has been helping me in these research projects. His tremendous efforts are essential to my growth as a researcher. I will never forget the nights when we worked side by side and slept on the floor of the lab. Thanks to all ADAPTers in our lab for the wonderful research environment you built.

I would also like to thank my parents. Both of them were not lucky enough to even finish their middle school education when they were teenagers, but they have been always super supportive to my education since the first day I came to this world. Now, I am going to become a PhD student in Computer Science this fall, while I would say that they have been PhDs in "Nurturing Me" for so many years. They let me know what education can do to a person, with their sympathetic ear and reliable back.

Last but not least, thanks to my girlfriend, Xianying Qin, for being there all the time.

Thank you very much, everyone!

Bill Yuchen Lin

Chengdu, May 28, 2018