

Doctoral Thesis

**A Study on Resolution and Retrieval of Implicit Entity
References in Microblogs**

Jun-Li Lu

February 2020

Department of Social Informatics
Graduate School of Informatics
Kyoto University

Committee of Doctoral Thesis:

Katsumi Tanaka, Professor

Masatoshi Yoshikawa, Professor

Keishi Tajima, Professor

Sadao Kurohashi, Professor

A Study on Resolution and Retrieval of Implicit Entity References in Microblogs

Jun-Li Lu

Abstract

In the age of social media, people share their opinions about the real world by writing microblogs in a SNS platform (e.g., Twitter/Facebook) to refer to entities (e.g., people/objects) and events (e.g., news), and analyzing those references in microblogs can be useful for valuable applications, such as targeted advertising and a low-cost way for public opinion survey. Meanwhile, the literature has extensively studied explicit entity references in microblogs, based on their literal contents (e.g., names or keywords). However, the implicit reference cases, which are not based on literal contents of entities, are increasing in SNS platforms and not yet fully studied for certain types of implicit references.

The difficulty of tackling implicit references can be explained from the following two typical types of implicit references, according to our survey results: (I) SNS users often mentioned entities (e.g., people/objects) without using their full/partial name or abbreviation in the contents of microblogs and (II) SNS users often posted microblogs to refer to events (e.g., news) without using event contents. For example, a user posted “Jacoby Ellsbury is leaving for the rival” and a reader is difficult to understand what “the rival” is if he/she does not know the fact that “Jacoby Ellsbury, a baseball player who belonged to Boston Red Sox, was being traded to New York Yankees, a rival of Boston Red Sox”. We therefore study on resolution and retrieval of implicit entity references in microblogs, as follows.

(1) To identify multiple indirectly or directly referred entities in microblogs, we propose effective features, such as topic-based semantics and syntactic patterns, which are built upon characteristics of indirect references. We propose Conditional Random Field (CRF) model with adaptive dependency to find indirectly referred entities. We build accurate and computationally efficient dependency.

(2) To identify implicitly or explicitly referred events from microblogs, we utilize reference-type-based event contents generation. To generate contents for implicitly

referred events, we mine similar links among microblogs and events by clustering them respectively. We also predict reference type to identify implicitly or explicitly referred events.

(3) Since microblogs and events are from heterogeneous spaces (i.e., opinions/feelings in microblogs vs. knowledge in news), we find meaningful similarity between microblogs and implicitly or explicitly referred events by dually transforming microblogs and events respectively. We therefore propose a deep transfer learning framework with dual transformations (DT). To identify implicitly referred events, we propose reference-type-based DT and apply DT with dense modeling that contains more layers of neural networks.

(4) We search for microblogs implicitly or explicitly referring to events, through search engine. The challenging is that searching for microblogs implicitly referring to events is a low-recall task, the search results returned by search engine are limited, and we try to maximize search efficiency for a fixed amount of queries, through search engine. To this end, we propose a search process that takes dynamic query strategies optimized by search experiences, with the following components. (a) To search for microblogs implicitly or explicitly referring to events, we propose exploratory and exploitative query strategies. (b) To accurately monitor the search process, we propose a robust reinforcement learning framework, which comprises deep Q-learning with RNN model for predicting long-term returns of search process by using sequential search states. (c) To increase the applicability of this research, i.e., the search process can be applied to unseen or new events, we optimize online query strategies by estimating the relevance of microblogs using research (3).

We draw a brief conclusion of this study as follows. (I) We studied on resolution and retrieval of implicit entity references in microblogs. (II) We proposed effective models and features for identification of implicit entity references. (III) We proposed a robust and highly applicable model for microblog search, especially effective for implicit references and the target of unseen events.

Keywords: Implicit Entity References, Entity Identification, Microblog Search, Deep Transfer Learning, Deep Reinforcement Learning

CONTENTS

1 . Introduction	1
1.1 Background	1
1.2 Approaches	5
1.3 Thesis Structure	7
2 . Related Work	9
2.1 Common NLP Approaches	9
2.2 Transfer Learning for Relevance Estimation	12
2.3 Reinforcement Learning for Focused Crawling	12
3 . Entity Identification by CRF Model with Adaptive Dependency	13
3.1 Introduction	13
3.1.1 Motivation	16
3.1.2 Novelty of Our Approach	17
3.2 Problem Definition	18
3.3 Features for Indirect or Direct References	19
3.3.1 Topic-based Semantics	20
3.3.2 Patterns of Phrases	21
3.4 CRF Model with Adaptive Dependency	22
3.4.1 Candidate Entity Extraction	22
3.4.2 Accurate and Computationally Efficient Dependency	23
3.4.3 Training CRF Model by Structured SVM	25
3.5 Experimental Setting	26

3.5.1 Microblog Annotation	26
3.5.2 Comparative Methods	27
3.6 Experimental Evaluation	30
3.6.1 Performance of Candidate Entity Extraction	30
3.6.2 Overall Performance	32
3.6.3 Feature Comparison	33
3.6.4 Model Performance	34
3.6.5 Case Study	35
3.7 Discussion	36
4 . Event Identification for Implicit and Explicit References	37
4.1 Introduction	37
4.1.1 Motivation	40
4.1.2 Novelty of Our Approach	41
4.2 Problem Definition	42
4.3 Reference-Type-based Event Contents Generation	43
4.3.1 Predicting Reference Type	43
4.3.2 Finding Similar Links by Clustering Microblogs and Events Respectively	44
4.3.3 Generating Event Contents for Implicit or Explicit References	46
4.4 Experimental Setting	48
4.4.1 Microblog Annotation	48
4.4.2 Comparative Methods	50
4.5 Experimental Evaluation	51
4.5.1 Performance on Implicit or Explicit References	51
4.5.2 Performance of Reference Type Prediction	54
4.5.3 Effect of Clustering	54
4.5.4 Case Study	55
4.6 Discussion	56

5 . Event Identification by Deep Transfer Learning with Dual Transformations...	57
5.1 Introduction	57
5.1.1 Motivation and Novelty of Our Approach	60
5.2 Problem Definition	62
5.3 Deep Transfer Learning with Dual Transformations	63
5.3.1 Embedding Vectors for Microblogs and Events Respectively	64
5.3.2 Dual Transformations for Microblogs and Events Respectively	65
5.3.3 Applying DT with Dense Modeling or with CNN Model	66
5.3.4 Reference-Type-based DT	66
5.4 Experimental Setting	67
5.4.1 Dataset Collection	68
5.4.2 Comparative Methods	70
5.4.3 Evaluation by Unseen Events	72
5.4.4 Space Dimension after Dual Transformations	73
5.5 Experimental Evaluation	74
5.5.1 Performance of Dual Transformations (DT)	77
5.5.2 Performance of DT with Dense Modeling or with CNN Model	80
5.5.3 Performance of Reference-Type-based DT	82
5.5.4 Visualization of How Dual Transformations Worked	83
5.6 Discussion	84
6 . Searching for Microblogs Referring to Events by Deep Dynamic Query Strategies	85
6.1 Introduction	85
6.1.1 Motivation	89
6.1.2 Novelty of Our Approach	91
6.2 Problem Definition	93
6.3 Search by Exploratory and Exploitative Query Strategies	94

6.3.1 MDP-based Search Process	94
6.3.2 Exploratory and Exploitative Query Strategies	97
6.4 Deep Q-learning with RNN Model	98
6.4.1 Predicting Long-Term Returns by RNN Model	100
6.5 Optimizing Online Query Strategies	102
6.6 Experimental Setting	102
6.6.1 Dataset Collection	102
6.6.2 Search Engine and Query	103
6.6.3 Comparative Methods	104
6.6.4 Evaluation by Unseen Events	107
6.6.5 Tuning of Deep Dynamic Search Process	108
6.7 Experimental Evaluation	112
6.7.1 Search Performance with Varying Relevant Microblogs	112
6.7.2 Effect of Optimizing Online Query Strategies	114
6.7.3 Effect of Deep Q-learning with RNN Model	115
6.7.4 Effect of Exploratory Query Strategies	118
6.7.5 Robust to Ranking Criteria of Search Engine	118
6.7.6 Case Study	120
6.8 Discussion	122
7 . Conclusions	124
7.1 Summary	124
7.2 Future Directions	126
Acknowledgements	128
Bibliography	129
Publications	137

LIST OF FIGURES

Figure 1.1: An implicit entity reference in a microblog	2
Figure 1.2: Overview of our approaches.....	4
Figure 3.1: Adaptive CRF model to identify indirectly referred entities.....	15
Figure 3.2: Performance of candidate entity extraction.....	30
Figure 3.3: Overall performance on models and features.....	31
Figure 3.4: Feature effect on indirect or direct references.....	31
Figure 3.5: Feature comparison	31
Figure 3.6: Performance of best feature combination.....	32
Figure 3.7: Model comparison.....	32
Figure 3.8: Performance of adaptive CRF and other CRFs.....	32
Figure 4.1: Reference-type-based method to identify implicitly referred events	38
Figure 4.2: Generating event contents for implicit or explicit references	43
Figure 4.3: Performance on our proposed and baseline method.....	52
Figure 4.4: Performance of clustering	53
Figure 5.1: Finding meaningful similarity between microblogs and implicitly referred events by dual transformations	58
Figure 5.2: Deep transfer learning with dual transformations	63
Figure 5.3: Word-sequence mapping from microblog to event.....	70
Figure 5.4: Single-direction transformations for microblogs	71
Figure 5.5: Tuning of space dimension after transformations	73
Figure 5.6: Performance of event identification	75
Figure 5.7: Performance of event identification on large-scale data	76
Figure 5.8: Performance of dual transformations with dense modeling.....	78
Figure 5.9: Performance of dual transformations with CNN model.....	79
Figure 5.10: Performance of reference-type-based DT	81
Figure 5.11: Visualization of how dual transformations worked	83

Figure 6.1: Search by robust reinforcement learning with exploratory and exploitative query strategies	86
Figure 6.2: Exploratory and exploitative query strategies to search for microblogs implicitly or explicitly referring to events	87
Figure 6.3: Predicts long-term returns by RNN model with sequential search states	98
Figure 6.4: Predicts long-term returns by only current search state	104
Figure 6.5: Pattern query strategies	105
Figure 6.6: Microblog search performance under varying relevant microblogs	111
Figure 6.7: Effectiveness of optimizing online query strategies.....	114
Figure 6.8: Diverse microblogs were found by exploratory and exploitative query strategies	117
Figure 6.9: Our model was robust to ranking criteria of search engine.....	119

LIST OF TABLES

Table 1.1: Our problems and approaches with past researches	4
Table 3.1: Entity identification problem and related works	16
Table 3.2: Comparison of modeling dependency	16
Table 3.3: Data summary	27
Table 3.4: Annotation results on tweets.....	27
Table 3.5: Features.....	28
Table 3.6: Parameters of experiments.....	29
Table 3.7: Case study of entity identification	33
Table 4.1: Event identification problem and related works.....	40
Table 4.2: Comparison on features for referred events.....	40
Table 4.3: Data summary	48
Table 4.4: Annotation results on tweets	48
Table 4.5: Parameters of experiments.....	50
Table 4.6: Performance of reference type prediction.....	53
Table 4.7: Case study on implicitly referred events	54
Table 5.1: Comparison on probabilistic modeling for event identification	60
Table 5.2: Transformation approaches	61
Table 5.3: Microblog statistics.....	67
Table 5.4: Event (news) annotation	67
Table 5.5: Preprocess of microblogs' locations.....	69
Table 5.6: The dataset setting for evaluation by unseen events.....	72
Table 6.1: Microblog search problem and related works	89
Table 6.2: Comparison on reinforcement learning frameworks	91
Table 6.3: The dataset setting for evaluation by unseen events.....	107
Table 6.4: Parameters of our models	108
Table 6.5: Tuning of predicting long-term returns by RNN model.....	108
Table 6.6: Microblog search performance	110

Table 6.7: Microblog search performance under varying relevant microblogs.....	110
Table 6.8: Effect of exploratory query strategies.....	118
Table 6.9: Case study of microblog search.....	121

CHAPTER 1

1. INTRODUCTION

1.1 Background

In the age of social media, people can share their opinions about the real world (e.g., news/celebrity) by writing microblogs in SNS platforms (e.g., Twitter/Facebook). Microblogs contain local opinions and in-house information that do not exist in public information, such as news articles and official pages of celebrity. Mining microblog data is useful for multiple valuable applications, such as targeted advertising and a low-cost way for the survey of public opinions.

Identifying explicit entity references in the literature

One important research topic in mining microblogs is to identify entities (e.g., people/objects) in textual contents of microblogs, and to identify referred events (e.g., news) from microblogs. For entity identification, the literature [37, 41-49, 107-109] has extensively studied the types of direct references, which are based on names of entities (i.e., full, partial, or abbreviation names). For event identification, most of previous works [1-20, 23-29, 110-115] identified events that are explicitly referred by microblogs using the contents of events (e.g., keywords).

1. Introduction



Figure 1.1: An implicit entity reference in a microblog. People used the text “the rival” to implicitly refer to entity “New York Yankees”.

Difficulty to identify implicit entity references

However, the cases of implicit entity references, which are not based on literal contents of entities, are frequently appearing in general SNS platforms and not yet fully studied. Our survey results suggested that 64.3% of tweets (microblogs) referred to news without containing news contents.

The difficulty of identifying implicit entity references can be explained as follows. A user posted “Jacoby Ellsbury is leaving for the rival” and a reader is difficult to understand the implicit reference of “the rival”, if he/she does not know the fact that “Jacoby Ellsbury, a baseball player who belonged to Boston Red Sox, was being traded to New York Yankees, a rival of Boston Red Sox”.

In implicit references, microblogs (or texts) referring to an entity can be diverse and implicitly referred entities can also be diverse. Therefore, common approaches such as keywords or co-occurrence may not be useful. For example, text-mentions implicitly referring to entities may be infrequent terms, or text-mentions and implicitly referred entities may be less co-occurred terms in documents.

1. Introduction

Searching for explicit entity references in the literature

Another important research topic is to search for event information from documents. Most of previous studies [64, 65, 67-70, 73, 74] searched for microblogs (documents) explicitly referring to events, by using event contents. Some studies [7, 25, 69, 70] searched for local or real-time microblogs. However, the extracted results can be limited by event contents or specifically spatial or temporal domains. We therefore search for microblogs implicitly referring to events without containing event contents, and thus we can find more diverse microblogs referring to events. To increase searchable microblogs, we search through search engine that contains dynamic or latest microblogs, compared with the static dataset used in previous studies.

Difficulty to search for implicit entity references, through search engine

Let us consider an example when a user tries to search for microblogs implicitly referring to a news. The user utilizes keywords of news but cannot find relevant microblogs without containing news contents. Then the user tries other queries based on his/her knowledge, it may not work because microblogs implicitly referring to events can come from different regions or writers, which the user may not be aware of.

The difficulty is as follows. There can be much less microblogs implicitly referring to events, e.g., below 1% relevant microblogs in our study. The search results provided by search engine are limited. For example, Twitter API limits the return results of a query and the number of issuable queries within time. The ranking criterion of search engine is also unknown to searchers. For example, even a user makes queries for textual contents, he/she may not get relevant microblogs that are posted in the past time or specific regions because the search engine prefers the latest or global search results.

1. Introduction

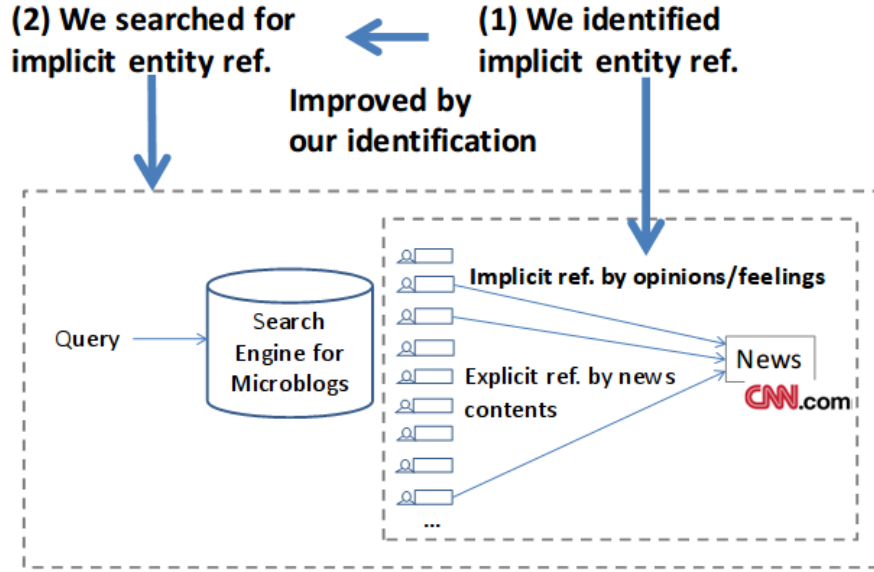


Figure 1.2: Overview of our approaches for identification and retrieval of implicit entity references in microblogs.

Table 1.1: Comparison of our problems and approaches with past works.

	Identification/Retrieval problem	Models/Features	
Implicit/ Explicit references	- Identify or search for implicitly/explicitly referred entities - Search for diverse microblogs w/o temporal or spatial limits	- Reference-type-based - Adaptive dependency - Deep transfer learning with dual transformations - Deep reinforcement learning with exploratory and exploitative query strategies	Our work
Explicit reference	- Named entity disambiguation - Identify or search for real-time or local events	- Keyword, co-occurrence, temporal, spatial - Linear/predefined dependency - Deep learning - Reinforcement learning	Past works

1.2 Approaches

Most of previous studies identified or searched for explicit entity references in microblogs or documents. In this study, we therefore focus on identifying or searching for implicit entity references in microblogs, as the following research topics.

(1) Identifying multiple indirectly or directly referred entities in a microblog

By observing writing styles of referring to entities in microblogs, we propose features that are effective for indirectly referred entities, such as topic-based semantics and syntactic patterns. For example, if we know that “baseball” is the topic of a microblog “Jacoby is the best player in the world!”, we may recognize “player” as “outfielder”, based on the topic of baseball. To find indirectly referred entities, we propose Conditional Random Field (CRF) with adaptive dependency. We dynamically build accurate and computationally efficient dependency among referred entities in a microblog.

(2) Identifying implicitly or explicitly referred events from microblogs

We propose two approaches for event identification. (a) By observing that referred events can be much different according to reference types, we use reference-type-based event contents generation for event identification. To generate contents for implicitly referred events, we mine similar links among microblogs and events by clustering them respectively. Note that these links are able to connect opinions or feelings in microblogs to events. We predict reference type to identify implicitly or explicitly referred events. (b) Since microblogs and events are from heterogeneous spaces (i.e., opinions/feelings in microblogs vs. knowledge in news), we find meaningful similarity between microblogs and implicitly or explicitly referred events by proposing a deep transfer learning framework with dual transformations (DT). In DT, we transform microblogs and events respectively such that the transformed microblogs and events are easier to be matched. For effective transformations, we utilize all features of microblogs and events. In our experiments, we use textual contents, writer profiles, posted locations, and time of microblogs or events. To identify implicitly referred events, we apply DT with competitive models. We propose reference-type-based DT and apply DT with dense modeling that contains more layers of neural networks or with CNN-based models.

(3) Searching for microblogs implicitly or explicitly referring to events, through search engine

The challenging is that searching for implicit references is a low-recall task, the search results returned by search engine are limited, and we try to maximize search efficiency for a fixed amount of queries, through search engine. To this end, we propose a search process that takes dynamic query strategies optimized by sequential search experiences with the following components. (a) To search for microblogs implicitly or explicitly referring to events, we propose exploratory and exploitative query strategies. (b) To accurately monitor the search process, we propose a robust reinforcement learning framework, which comprises deep Q-learning with RNN model for predicting long-term returns of search process by using sequential search states and query strategies. (c) To increase the applicability of this research, i.e., the search process can be applied to not only events seen during training phase but also to unseen or new events, we optimize online query strategies by estimating the relevance of microblogs using the model proposed in the above (2).

1.3 Thesis Structure

The remainder of this thesis is organized as follows.

- Chapter 2
We discuss and overview previous studies related to our research topics presented in this thesis.
- Chapter 3
We address the problem of identifying multiple indirectly or directly referred entities in a microblog. We especially tackle indirect references by developing features, such as topic-based semantics. To find indirectly referred entities, we model Conditional Random Field (CRF) model with adaptive dependency. Experimental results showed the effectiveness of adaptive CRF model and our features for certain types of indirect references.
- Chapter 4
We address the problem of identifying implicitly or explicitly referred events from microblogs. To this end, we utilize reference-type-based event contents generation. To generate contents for implicitly referred events, we mine similar links among microblogs and events by clustering them respectively. We also predict reference type to identify implicitly or explicitly referred events.
- Chapter 5
We propose a novel approach for event identification. Since microblogs and events are different in their contents, we find meaningful similarity between microblogs and implicitly or explicitly referred events by proposing a deep transfer learning framework with dual transformations (DT). For effective transformations, we utilize all features that include textual contents, writer profiles, posted location and time of microblogs or events. To identify implicitly referred events, we propose reference-type-based DT and apply DT with dense modeling that contains more layers of neural networks. In experiments, our models could be applied to unseen events and were effective on both references and unseen events.
- Chapter 6
We address the problem of searching for microblogs implicitly or explicitly referring to events, through search engine. To this end, we propose a search

1. Introduction

process that takes dynamic query strategies optimized by sequential search experiences, with the following components. (a) We propose exploratory and exploitative query strategies to search for microblogs implicitly or explicitly referring to events. (b) To accurately monitor the search process, we propose a robust reinforcement learning framework that predicts long-term returns of search process by RNN model using sequential search experiences. (c) To increase the applicability of this research, i.e., the search process can be applied to unseen or new events, we optimize online query strategies by estimating the relevance of microblogs using the estimator proposed in Chapter 5. Experimental results showed the effectiveness of our method in searching for implicit or explicit references and the target of unseen events, and the robustness of our models.

- Chapter 7

We summarize the thesis and address several directions to be explored in future work.

CHAPTER 2

2. RELATED WORK

2.1 Common NLP Approaches

Entity identification

Entity identification has been an area of active research and attracted many researchers. There are two kinds of problems relating to entity identification. The first problem is to identify which part of words (denoted as a mention) in a document refers to an entity [40-44]. The second problem is to identify which entity is referred to by that mention [36, 37, 45-47]. Some studies addressed both problems together to identify referred entities by detected mentions at the same time [42]. Some tackled the problem on well-written documents [45], [48] and on short-and-noisy microblogs such as tweets [40-44] and some proposed time-and-space-efficient algorithms [49].

To evaluate referred entities, the features used in previous studies include the probability that mentions entity name [42, 50], the similarity between the context of entities and a document [45, 48, 51], the correlation between multiple entities [52-54], and user interest [47]. One proposed mining context in addition to the knowledge base [48].

The drawback of previous studies is as follows. Most of related works identified only the referred entities by using full/partial/abbreviation names. However, their approaches might not be effective for entities referred without using names. E.g., the text “the president” is used to refer to the entity “Barack Obama” and the text is not entity name.

2. Related Work

In addition, the text of the mention is often normal nouns, which can be abstract terms, and thus the candidate entities can be much different and in diverse topics. This implies that we should consider more clues for identification, such as more features among mentions and referred entities, or more accurate dependency among multiple referred entities. We therefore identify indirectly referred entities in Chapter 3.

Event identification

Event identification or detection has been an area of active research and attracted many researchers. Given documents, one task is to detect whether documents are referred to events or not, and another task to identify which events are referred to by documents [1, 2, 4-6]. We categorize studies related to event identification or detection as follows. Some studies extracted real-time events from microblogs (e.g., tweets) of a social media platform for providing up-to-date information about the real world [7, 8]. Some studies analyzed the detected events from historical microblogs across a time period for understanding behaviors of people mentioning events of past or expectation to future [9, 10]. Some studies searched the first-story of events from microblogs for extracting the reason or origin of events [3, 11, 12]. Some studies summarized an amount of microblogs across time periods and geographical locations for providing an overview or visualization of events with respect to time and locations [10, 13-18]. Some studies acquired the information related to events by extracting geospatial location mentions or opinions, sentiments, or actions from microblogs [4, 19, 20, 21]. Some studies detected events from news articles [22] and some studies compared behaviors of references of events between a social media platform and a traditional news platform [23, 24]. For the above studies, the generation of candidate events was usually according to event contents.

To evaluate referred events, the features used in previous studies are listed as follows. Microblog features [8, 25]: the studies used textual contents including symbols (e.g., hashtag or emotional notations) and hyperlinks, posted locations and time of a microblog, and replied microblogs. Social media platform features [7, 8, 24, 26-28]: the studies used key or trending terms, which are suddenly frequently appearing terms or frequently queried terms on a social network platform, and modeled topics and spatial or temporal aspects of a set of microblogs. Knowledge base features [8, 26, 29]: the studies used additional knowledge such as information of entities, which could be extracted from Wikipedia or web search. NLP (natural language processing) features [2, 3, 11, 15]: the

2. Related Work

studies used the mining results of documents by NLP techniques such as importance, topics, semantics, and correlation among a set of terms, and basic parsing including part of-speech (e.g., noun) or functional terms (e.g., question words).

The drawback of previous studies is as follows. However, in microblogs, people can refer to events by various ways, such as using their opinions or feelings. In these kinds of implicit references, microblogs do not contain event contents and thus the candidate events can be much different and in diverse topics. Therefore, keyword-based or co-occurrence-based features might not be useful and we should consider more appropriate features or probabilistic modeling. We therefore identify implicitly referred events in Chapters 4 and 5.

Crawling for event information

Crawling for event information from multiple data sources, such as documents, web pages, and microblogs, has been extensively studied. These studies [64-68] utilized multiple features, including keywords or named entities of textual contents, corresponding spatial locations or temporal information, the information of writers, and the correlation between the target and crawled data. In some studies [68], the credibility or quality of the extracted event information was improved by evaluating data source or metadata (e.g., web media platforms of data source or the initial seeds of data), or mining typical patterns of data. Other studies [69, 70] provided event information in real time by extracting up-to-date information from social media platforms. For example, the proposed methods efficiently identify the typical patterns or unusual bursty streams in multiple contexts (text, locations, or time) of microblogs.

However, the crawled event information from previous studies can be limited by keywords or spatial or temporal domains. In contrast, we find more diverse information about events without spatial or temporal limitation. We thus search for microblogs implicitly or explicitly referring to events, through search engine, in Chapter 6.

2.2 Transfer Learning for Relevance Estimation

To identify relevant microblogs for target events, especially for those events that do not exist in training data, we utilize transfer learning [79]. In approaches of transfer learning, zero-shot learning (ZSL) [78] is a kind of problem setting that a classifier can categorize instances into classes where their instances do not exist in training data. ZSL has been applied to the learning of intermediate attribute classifiers [80-83], the learning for mixture of seen class proportions [84-86], and the learning of the compatibility among heterogeneous spaces [87-89]. A typical ZSL framework [89] links the unseen images to the correct text labels by mapping the image representations in convolutional neural network (CNN) models to the text representations in Word2vec models.

In their modeling, they often utilized single-direction transformations that transforms from the space of instances into the space of classes. However, in our problem, both of microblogs and their referred events can be much different (opinions/feelings in microblogs vs. knowledge in news events), and thus we consider to dually transform for both sides of microblogs and events, within a deep transfer learning framework, in Chapter 5.

2.3 Reinforcement Learning for Focused Crawling

Reinforcement learning frameworks can be used for modeling behaviors of exploration in the problem of focused crawling [71-74] that searches for relevant documents or records in a set of data, or the problem of playing Atari games [75-77, 116]. They often utilized limited or predefined actions in reinforcement learning modeling. For example, they utilized exploitative types of queries [71, 73], such as keyword-based queries, which can be useful to search for typically relevant documents. However, for our low-recall search problem, we are required to search for more diverse or unseen microblogs (documents), and thus exploitative types of queries might not be useful. We therefore propose to use exploratory types of queries, within a robust reinforcement learning framework in Chapter 6.

CHAPTER 3

3. ENTITY IDENTIFICATION BY CRF MODEL WITH ADAPTIVE DEPENDENCY

3.1 Introduction

People write microblogs to share their opinions about information in the real world and refer to entities, such as people and places, in various ways. Entity identification, a problem of identifying entities referred to in a document, has been extensively addressed in the literature [36, 37]. Most related studies addressed the problem of identifying entities referred to by their full/partial name or abbreviation. In microblogs, however, there are many entities mentioned in an indirect way, which are often difficult to identify, especially in short text such as microblogs. Given a microblog “Jacoby Ellsbury is leaving for the rival”, e.g., it must be difficult to understand what “the rival” is if one does not know the fact that “Jacoby Ellsbury, a baseball player who belonged to Boston Red Sox, was being traded to New York Yankees, a rival of Boston Red Sox”.

In this paper, we address the entity identification problem with special attention to indirect reference cases in which entities are not referred to by their names. Compared with direct reference cases, where entities are referred to by their names, indirect references usually have more candidate entities and are accordingly difficult to identify. We therefore consider more evidence for entity identification that is especially important

3. Entity Identification by CRF Model with Adaptive Dependency

for indirect references. E.g., we use syntactic patterns that are useful in identifying indirectly referred entities. If patterns, such as “Y known as X” and “X regarded as Y ”, appear many times, they can be strong cues that X is used to refer to Y and be only applicable to indirect reference cases. Another example of effective features is a set of words in a microblog that is translated into topic-based semantics. E.g., if we know that “baseball” is the topic of a microblog “Jacoby is the best player in the world!”, we may translate “player” to “outfielder” and recognize “Jacoby” as “Jacoby Ellsbury”.

We also use dependency among entities in a microblog for our entity identification problem. Since multiple entities can be mentioned in a microblog and be strong clues to identify other entities, predicting referred entities together is essential for accurate entity identification. We therefore use a Conditional Random Field (CRF) model [38, 39] to model the correlation among referred entities in a microblog. A limitation of linear CRF models is that they cannot explicitly represent cases in which distant entities are dependent or neighboring entities are independent. Thus, we also model the non-sequential order dependency while keeping the inference tractable by dynamically building dependency among entities.

Experiments were carried out with 461 tweets that were randomly collected from Twitter¹. The experimental results suggested that 1) our features were effective for indirect references, 2) our CRF model with adaptive dependency was robust even when there were multiple mentions in a microblog, and 3) it achieved the same high performance as the fully connected CRF model. We summarize the contributions of this work:

- We address the problem of identifying indirectly mentioned entities and argue that there is a considerable amount of indirect references in real microblog data.
- We propose features built upon characteristics of indirect references and a CRF model that dynamically builds dependency for identifying referred entities.
- We demonstrate that our features worked for both direct and indirect references and the CRF model with adaptive dependency is advantageous over several variants of CRF models.

¹ <https://twitter.com>

3. Entity Identification by CRF Model with Adaptive Dependency

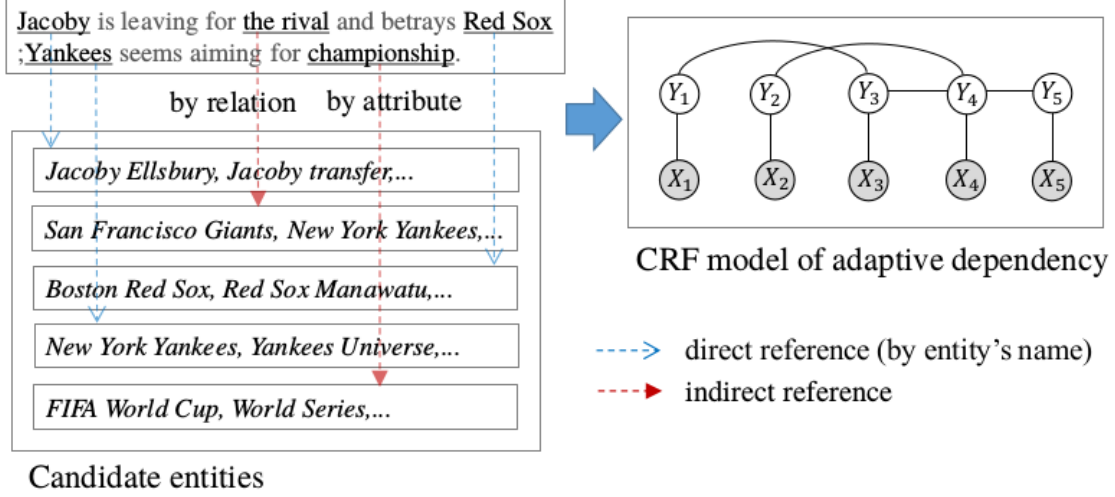


Figure 3.1: Entity identification on a microblog in the presence of direct/indirect references, where there are five mentions “Jacoby”, “the rival”, “Red Sox”, “Yankees”, and “championship” denoted as X_1 , X_2 , X_3 , X_4 , and X_5 and the corresponding referred entities denoted as Y_1 , Y_2 , Y_3 , Y_4 , and Y_5 , respectively.

The rest of this research is organized as follows. We show the motivation and novelty compared with previous studies in Section 3.1. The entity identification problem is presented in Section 3.2. We propose features and a CRF model with adaptive dependency in Sections 3.3 and 3.4. We evaluated the proposed methods using real-world microblog data, discussed the experimental results, and presented a case study in Sections 3.6 and 3.7.

3. Entity Identification by CRF Model with Adaptive Dependency

Table 3.1: Entity identification problem and related works.

	Problem: entity identification for indirect or direct references	Problem: Named Entity Disambiguation	Problem: Named Entity Recognition
Ours (Ch 3)	o	o	
[42, 45]		o	
[37, 46-48]		o	
[49]		o	
[41, 43-44, 107-109]			o

Table 3.2: Method comparison. We compare on modeling of dependency among multiple referred entities.

	Dependency among referred entities				
	Adaptive-dependency CRF	Fully connected dependency	Sequential dependency	Predefined dependency (by knowledge graph)	Fewer dependencies
Ours (Ch 3)	o				
[42, 45]		o			
[37, 46-48]				o	
[49]					o
[41, 43-44, 107-109]			o		

3.1.1 Motivation

Identifying multiple indirectly or directly referred entities in a microblog

We try to identify the entities that are not mentioned by their names, which can be difficult, compared with those previous works [37, 41-49, 107-109] that identified only entities mentioned by full/partial/abbreviation names. The difficulty is that the candidate entities of indirect references can be in diverse topics and be large in number, compared with those candidates mentioned based on their names. We showed that the number of candidates was in exponential level (i.e., 10^3) to make sure the ground-truth ones existing in the set of candidates. This implies that we should find more clues. Moreover, the dependency among indirectly or directly referred entities can be complicated due to the uncertainty of indirectly referred candidates, compared with the limited dependency induced by only directly referred entities.

3.1.2 Novelty of Our Approach

Accurate and computationally efficient dependency among multiple indirectly or directly referred entities

With the challenging induced by indirect references, we therefore utilize more and diverse features for identifying indirectly or directly referred entities. We propose features that are effective for certain types of indirect references. We advocate utilizing the adaptively non-cycle dependency among entities because we can capture accurate and computationally efficient dependency, compared with the modeling of dependency in previous works of Table 3.2.

The drawbacks of modeling dependency in previous works are as follows. In fully connected dependency [42, 45], the risk is inaccurate dependency and the exponential complexity of computation. Note that the complexity can be $O(c^k)$, where k is the length of the longest cycle, and the time is $O(nc^2)$ for our adaptive dependency. The drawback of sequential (linear) dependency [41, 43, 44, 107-109] is that sequential order, such as the appearing order in a microblog, cannot reflect proper dependency. We observed that the contents of microblogs were often less sequential because of free writing styles and the properties of short text. Furthermore, predefined or static dependency [37, 46-48] might be inappropriate because the writing contents can be much different according to different writers or motivations of the posting.

3.2 Problem Definition

We introduce several terms to define the problem addressed in this paper as shown in Figure 3.1. *Microblog*: a microblog is a sequence of words posted by a user on a social media platform, e.g., a tweet posted by a user on Twitter is a kind of microblog. *Mention*: a mention is a sequence of words in a microblog and is used by a user to refer to some entity, e.g., a person, object, or place, in the real world. Note that we assume a one-to-one relationship between an entity and a Wikipedia article; thus, an entity must correspond to a single Wikipedia article². E.g., there are five mentions, each of which has underlined words, in the microblog in Figure 3.1. We then describe how an entity can be mentioned in the contents of microblogs as follows.

Direct reference: Previous studies [36, 40] focused on entities that are mentioned based on their names. If mention m is used to refer to entity e and m is the full/partial name or the abbreviation of the name of e , the reference to e by m is a direct reference. Note that the name of e is defined by the title of the article of e in Wikipedia. E.g., in Figure 3.1, m is “Jacoby” and candidate entities e_1 and e_2 are “Jacoby Ellsbury” and “Jacoby transfer”, respectively, because m is the partial name of e_1 and that of e_2 . However, the scope of referred entities by direct reference is limited because people can refer to entities without using their names. Motivated by the research of analogy and metaphor [55–57] in linguistics, we target referred entities of the following novel type.

Indirect reference: if mention m is used to refer to entity e and m is neither the full/partial name nor the abbreviation of the name of e , the reference to e by m is an indirect reference. E.g., e = “Jacoby Ellsbury” can be indirectly referred to by each m of the following cases: 1) m is the attribute that describes e , e.g., m = “the player”. 2) m is the relationship between e and another entity e_2 , e.g., m = “outfielder”, where e is an outfielder of e_2 = “New York Yankees”. 3) By metaphor, m contains the name of another entity e_2 , e.g., m = “our Ichiro Suzuki”, where e_2 = “Ichiro Suzuki” and the referred e are both outfielders. Note that we avoid unrealistic references between a mention and an entity by limiting a mention to be a noun-phrase.

² <https://en.wikipedia.org/wiki>

3. Entity Identification by CRF Model with Adaptive Dependency

Problem Definition: (Entity identification on microblogs in the presence of indirect or direct references): given a microblog b from a user u , a set of mentions M_b extracted from b , and a set of entities E , the problem is to identify the entity $e \in E$, which is referred to by each mention $m \in M_b$, through indirect or direct reference.

3.3 Features for Indirect or Direct References

We use features to evaluate a direct or indirect reference between a mention and a referred entity. We especially focus on indirect reference cases and design features for them by comparing previous studies. Given a set of mentions M (denoted as M for convenience) from a microblog, for each candidate entity e of each mention $m \in M$ and for each pair of candidates e and e_2 of a given pair of mentions $m \in M$ and $m_2 \in M$, we use the following features. At first, we consider two basic features.

Keyword: we consider whether the keywords K_e , which describe entity e and are extracted from the main (the first- k) paragraphs and the titles of categories in the article of e in Wikipedia, can match mention m . E.g., “outfielder” is one keyword for “Jacoby Ellsbury”. Therefore,

$$\text{sim}(K_e), \text{sim}(K_e, \{m, m_2, n_{e_2}\}). \quad (3.1)$$

Note that K_e is the top- k words that are picked according to the weight $tf_i * idf_i$ of each given word w_i , where tf_i is the frequency of w_i in all the specified documents³ (or words-windows) and idf_i is the inverted document frequency of w_i ;

$$\text{sim}(A, B) = \frac{a * b}{\|a\| \|b\|} \quad (3.2)$$

is the similarity between the bag-of-words model of A and that of B , where the i -th element of a (or b) belonging to A (or B) is the weight $tf_i * idf_i$ of word w_i . We also examine the neighboring mention m_2 and the name n_{e_2} of the neighboring entity e_2 in the same microblog by the similarity $\text{sim}(K_e, \{m, m_2, n_{e_2}\})$.

³ A document is an article in Wikipedia.

3. Entity Identification by CRF Model with Adaptive Dependency

Occurrence frequency: we consider the co-occurrence among entities e and e_2 and mention m in documents as

$$\log r(n_e), \log r(n_e, m), \log r(n_e, n_{e_2}), \quad (3.3)$$

where $r(n_e, m)$ is the number of times the name n_e of e and m co-occur in documents.

The difficulty in an indirect reference is that we cannot acquire referred entities by the mention. The reason is that a mention in this case often consists of normal nouns, which can be abstract and difficult to match entity's information, e.g., the topic of entities. For instance, in the microblog "X is the best player", the mention "player" can refer to entities of a variety of topics, e.g., a referred entity is an "outfielder" in the topic of baseball or "goalkeeper" in the topic of football.

3.3.1 Topic-based Semantics

We therefore translate a microblog including mentions based on the related topics and use the translated content to accurately evaluate referred entities. As in the above example, we may translate the mention "player" based on the topic of the microblog, into either "outfielder" or "goalkeeper" in an ideal situation. The process of translating the content W_b of a microblog b is as follows.

1. We extract the top proper-nouns P_b , which are related to microblog b of writer u . The words of P_b are extracted from W_b , news linked by b , and descriptions of users tagged in b , and the recent microblogs written by u .
2. We then extract the words I_b from Wikipedia documents, which often co-occur with the words of P_b .
3. Using I_b , we translate word $w \in W_b$ by word $w' \in I_b$ if w and w' are semantically similar, i.e., $\{\{w' \mid \lambda(w, w') \geq \theta, w \in W_b, w' \in I_b, 0 \leq \theta \leq 1\} \cup W_b\}$. Note that stop-words and proper nouns in microblog b are not needed to be translated and we keep the weight of each translated word from I_b .

3. Entity Identification by CRF Model with Adaptive Dependency

The semantic similarity $\lambda(w, w')$ between words w and w' is measured by WordNet Similarity [58], which takes into account the lexicalized concept, synset, taxonomy, and dictionary definitions for all words and is provided by this tool⁴.

With the translated microblog T_b , we evaluate an entity e by computing the similarity between T_b and the context C_e of e , which is a set of words from each words-window surrounding the name n_e of e from documents, e.g., a words-window is “ n_e hits a home-run”. Therefore,

$$\text{sim}(C_e, T_b), \text{sim}(C_{ee_2}, T_b). \quad (3.4)$$

We also consider the context of e and the neighboring entity e_2 as C_{ee_2} . To consider the diversity of the context of entities, we separate the context C_e based on parts of speech into a set of nouns $C_{e,n}$ and a set of verbs $C_{e,v}$. E.g., the noun “outfielder” belongs to $C_{e,n}$ and the verb “hit” belongs to $C_{e,v}$. Therefore,

$$\text{sim}(C_{e,n}, T_b), \text{sim}(C_{e,v}, T_b), \text{sim}(C_{ee_2,n}, T_b), \text{sim}(C_{ee_2,v}, T_b). \quad (3.5)$$

3.3.2 Patterns of Phrases

Furthermore, we consider linguistic features to acquire more clues based on the following two reasons. First, armed with topic-based semantics, we can identify entities of a certain topic; however, we are still ambiguous about that how people think of entities. Especially, how a microblog’s writer describes about an entity. Second, for the entity identification problem, few studies went into detail about the following linguistic features we propose.

Entity-known-as pattern: according to a pattern p , where people can utilize p to link an entity e with some object, we check whether e can be represented by mention m . i.e., we check if the words in p appear in sentences containing the name n_e of e and m . E.g., “ $n_e \dots p \dots m$ ” is a sentence such that n_e appears and is followed by p and m , where n_e =“Jacobry Ellsbury”, p =“known as”, and m =“outfielder”. Therefore,

⁴ <https://code.google.com/p/ws4j>

3. Entity Identification by CRF Model with Adaptive Dependency

$$\log \sum_{p \in P} l("n_e \dots p \dots m"), \log \sum_{p \in P} l("n_e \dots p \dots m \dots n_{e_2}"), \quad (3.6)$$

where $l("n_e \dots p \dots m")$ denotes the number of times sentence " $n_e \dots p \dots m$ " appears in documents. We further consider whether a neighboring entity e_2 is involved in by sentence " $n_e \dots p \dots m \dots n_{e_2}$ ".

Entity's action: people could describe the action, which is performed by an entity e . Therefore, we measure whether e performed an action by checking if e is reasonable to co-occur with the verbs v_s (or v_o), where v_s (or v_o) is the verb of a sentence in microblog b such that the corresponding mention m of e is the subject (or the object), by

$$p(e|v_s) = \frac{l("e \dots v_s")}{\sum_{e'} l("e' \dots v_s")}, p(e|v_o) = \frac{l("v_o \dots e")}{\sum_{e'} l("v_o \dots e'")}, \quad (3.7)$$

where the subject in sentence " $e \dots v_s$ " is e and the object in sentence " $v_o \dots e$ " is e .

3.4 CRF Model with Adaptive Dependency

3.4.1 Candidate Entity Extraction

Given a mention of a microblog, we acquire candidate entities that are possibly referred to by the mention, as follows. Because an entity e is mentioned from a microblog b on a social media platform p , we argue that there are writer's other microblogs, other users' microblogs on platform p , or web pages (e.g., news articles) containing information related to e and these data are useful to search for candidate entities. Therefore, we use search queries Q that are extracted from different document collections mentioned above. The search queries are

$$Q = \{(w_1, \varphi_1), (w_2, \varphi_2), \dots, (w_i, \varphi_i)\}, \quad (3.8)$$

where φ_i is the weight of word w_i . The process for extracting Q is shown as follows.

At first, we extract the set of top words Δ with the highest tf-idf weights, from different document collections denoted by S_1 , S_2 , and S_3 . S_1 is the related writer's microblogs searched by mention m , S_2 is the related users' microblogs searched on platform p using microblog b 's hashtags (e.g., "#yankees" in Twitter), and S_3 is the related web pages searched on Web using m . Because proper-noun words could reflect about the topic of

3. Entity Identification by CRF Model with Adaptive Dependency

entity e , we keep only proper-noun words in S_1 , S_2 , and S_3 and add user's description on platform p from user-hashtags into S_1 and S_2 (e.g., “@RedSox” is a user-hashtag in Twitter). Second, because Δ is extracted from the documents of different domains (i.e., S_1 , S_2 , and S_3), we make the search query Q from the words of Δ with words of balanced weighting on different domains as follows.

1. At first, for the words of Δ , the weight $\varphi_{i,S}$ of each word w_i from each domain S (e.g., words of S_1 are from one domain) is normalized as $\varphi'_{i,S}$ (we normalized by the max weight).
2. We determine the importance of each domain S as the weight φ_S .
3. We sum up the weight of each word w_i from all domains as $\sum_{S \in \{S_1, S_2, S_3\}} \varphi_S * \varphi'_{i,S}$.

3.4.2 Accurate and Computationally Efficient Dependency

Because there are multiple mentions in a microblog, we want to predict the referred entities of these mentions together by considering their correlation. We therefore formulate the problem by using a Conditional Random Field (CRF) model with the advantage of relaxing the independence assumption, which is made by using a Hidden Markov Model (HMM), and avoids the label bias problem [38, 39]. Furthermore, for making the inference that predicts referred entities on a CRF model tractable, we use a CRF model with non-cycle dependency, which does not contain cycle connections of dependency among referred entities. That is, the time complexity of the inference on the non-cycle CRF model is $O(nc^2)$, which is linear to the number n of mentions in a microblog, where c is the number of candidate entities of a mention; however, this complexity on a cycle-connected CRF model can be exponential to n , i.e., $O(c^k)$, where k is the length of the longest cycle and $3 \leq k \leq n$.

Given a set of mentions M from a microblog b , we convert it to a CRF model (X, Y, L) as follows. Each mention $m_i \in M$, $1 \leq i \leq |M|$ is denoted as X_i as an observed value, which is connected to a label Y_i . Y_i is a random variable and the value of Y_i is a candidate of the referred entity of m_i . Note that microblog b is denoted as X_0 . Formally $X = X_0 \times X_1 \times \dots \times X_n$, $Y = Y_1 \times \dots \times Y_n$, and L is a set of undirected connections of dependency among labels in Y . One example of formulating the problem into a non-cycle CRF model is shown in Figure 3.1. Given a set of mentions from a microblog denoted as $\mathbf{x} = (x_0, x_1, \dots, x_n) \in X$ the probability of a candidate of referred entities $\mathbf{y} = (y_1, \dots, y_n) \in Y$

3. Entity Identification by CRF Model with Adaptive Dependency

$$p(\mathbf{y}|\mathbf{x}) = \frac{\exp(F(\mathbf{y},\mathbf{x}))}{\sum_{\mathbf{y}'} \exp(F(\mathbf{y}',\mathbf{x}))}, \quad (3.9)$$

$$F(\mathbf{y}, \mathbf{x}) = \sum_{i=1}^n \left[\sum_{f \in F_\alpha} w_f f(y_i) + \sum_{f \in F_\beta} w_f f(y_i, x_i) + \sum_{(i,j) \in L} \sum_{f \in F_\gamma} w_f f(y_i, y_j) \right]$$

where F_α , F_β , and F_γ are a set of features of a single entity, a set of features between a mention and an entity, and a set of features between two entities, respectively, w_f is the weight of feature f , $(i, j) \in L$ denotes that label Y_i is connected to label Y_j , and $z(x)$ is the normalizing constant. The best candidate of referred entities ranked using the CRF model is

$$\mathbf{y}' = \operatorname{argmax}_{\mathbf{y} \in Y} p(\mathbf{y}|\mathbf{x}). \quad (3.10)$$

Given a microblog, we observed that some mentions in a microblog may come from different topics, which implies that dependencies among the corresponding referred entities are different. To enhance the inference on a CRF model, we acquire dependency among referred entities, i.e., to determine and give the weight to any connection between labels in Y in the CRF model. E.g., given a set of mentions from a microblog denoted as $\mathbf{x} = (x_0, x_1, \dots, x_4) \in X$, suppose mentions x_1 and x_2 are from the topic of “baseball” and those denoted as x_3 and x_4 are from the topic of “education”, the connections between Y_1 and Y_2 or between Y_3 and Y_4 are more dependent than those between Y_1 and Y_3 or between Y_1 and Y_4 .

We propose a greedy algorithm to dynamically acquire dependency among referred entities on a given microblog as follows. Given a CRF model of $(X, Y, L = \emptyset)$ from a microblog, we sequentially add one important connection $l = (i, j) \in U$ of the dependency between labels Y_i and Y_j into L from unselected connections in U , where l does not induce a cycle with existing connections in L , until there are no more important connections in U . The connection l is selected by

$$\operatorname{argmax}_{l \in U} \delta(l) \quad (3.11)$$

and we filter out l with the extreme-low-importance if $\delta(l) \leq \rho\tau$, where $\rho\tau$ is the threshold of importance value, τ is the mean of importance values of all connections, and $\delta(l)$ is the importance of $l = (i, j)$ between labels Y_i and Y_j that we measure by summing the top-k values of

3. Entity Identification by CRF Model with Adaptive Dependency

$$\sum_{f \in F_\beta} w_f f(y_i, x_i) \sum_{f \in F_\beta} w_f f(y_j, x_j) \sum_{f \in F_\gamma} w_f f(y_i, y_j), \quad (3.12)$$

where $\forall y_i \in Y_i, \forall y_j \in Y_j$, and each weight w_f is acquired by training. Note that the importance $\delta(l)$ can be measured by different ways. With importance values of connections, the probability of a candidate of entities is now defined by

$$F(\mathbf{y}, \mathbf{x}) = \sum_{i=1}^n [\sum_{f \in F_\alpha} w_f f(y_i) + \sum_{f \in F_\beta} w_f f(y_i, x_i) + \sum_{l=(i,j) \in L} \sum_{f \in F_\gamma} \delta'(l) w_f f(y_i, y_j)],$$

$$\delta'(l) = \frac{\delta(l)}{\sum_{l' \in L} \delta(l')}. \quad (3.13)$$

Proof (inference complexity on a non-cycle CRF model). Given a CRF model, which does not contain cycle connections of dependency, comprises a set of trees S , where each pair of trees in S is not connected, the inference on each tree $s \in S$ by the Viterbi algorithm takes time $O(n_s c^2)$ [63] and then the overall time is $\sum_{s \in S} O(n_s c^2) = O(c^2) \sum_{s \in S} n_s = O(nc^2)$, where $\sum_{s \in S} n_s = n$.

Inferring by Gibbs sampling. As the number of candidate entities for entity identification could be large, we efficiently picked the top-k candidates C with high score by Gibbs sampling. Given candidate $\mathbf{y}^{(i)}$ sampled at the current round i , we sampled candidate $\mathbf{y}^{(i+1)} = (y_1^{(i+1)}, \dots, y_n^{(i+1)})$ at the next round $i + 1$ as follows:

$$\begin{aligned} \text{sample } y_1^{(i+1)} &\sim p(y_1 | y_2^{(i)}, \dots, y_n^{(i)}), \\ &\dots \\ \text{sample } y_n^{(i+1)} &\sim p(y_n | y_1^{(i+1)}, \dots, y_{n-1}^{(i+1)}). \end{aligned} \quad (3.14)$$

3.4.3 Training CRF Model by Structured SVM

We train the CRF model by using Structured SVM (SSVM) with the objective function of Hinge loss [59], [60],

$$[\max_{\mathbf{y} \in Y} (\Delta(\mathbf{y}, \mathbf{y}^i) + F(\mathbf{y}, \mathbf{x}^i; \mathbf{W})) - F(\mathbf{y}^i, \mathbf{x}^i; \mathbf{W})]_+, \quad (3.15)$$

3. Entity Identification by CRF Model with Adaptive Dependency

where $\Delta(\mathbf{y}, \mathbf{y}^i) = \sum_{j=1}^n |\{1 | y_j \neq y_j^i\}|$ is the hamming distance and $\Delta(\mathbf{y}, \mathbf{y}^i) = 0$ if $\mathbf{y} = \mathbf{y}^i$. By training, we make the compatibility score of the relevant paired entities and mentions larger than that score of other paired entities. Meanwhile, it encourages the score of relevance $F(\mathbf{y}^i, \mathbf{x}^i)$ much larger than other $F(\mathbf{y}, \mathbf{x}^i)$ when the two given entities y and y^i are much different, i.e., $\Delta(\mathbf{y}, \mathbf{y}^i)$ is large. Note that we use the tool [62] for running SSVM training.

3.5 Experimental Setting

We explained the experimental setting for entity identification including the dataset collection, the annotation of microblog data, and comparative methods.

3.5.1 Microblog Annotation

We investigated 461 tweets (which were microblogs) from Twitter, which were randomly collected using five Twitter tags, as shown in Table 3.4, during October 22-27, 2014. Then, three people annotated these tweets according to the following three rules. Note that one of them was an author of this paper.

- A mention is a noun-phrase because words of other parts of speech are difficult to refer to entities. E.g., the verb “run” is obviously not used to refer to entities.
- Entities that are specific should be targeted. E.g., entity e = “Jacoby Ellsbury” is specific and entity e_2 = “Outfielder” is general. The reason is that if an entity (e.g., e_2) is not specific, there exists other entities replaceable for e_2 .
- Annotations should not be nested. E.g., if the text “United State of America” in a microblog is annotated as a mention to refer to a certain entity e , then “America” in this text cannot be annotated to refer to the same e .

Because the boundary of a mention text in a short-and-noisy tweet was not clear and sometimes the referred entities for the same mention from different annotators were similar in their Wikipedia articles, e.g., articles of “Ebola virus” and “Ebola virus disease” are similar, we relaxed annotation results in the same tweets annotated by

3. Entity Identification by CRF Model with Adaptive Dependency

Table 3.3: Data summary.

Data	#
Tweets	461
Mentions	1202
Direct references	850
Indirect references	570
Writers	354
News tagged in tweets	375
Twitter-users tagged in tweets	210
Articles (entities) in Wikipedia (version 2014-12-08)	4,732,221

Table 3.4: Annotation results on 461 randomly collected tweets.

Twitter-tag	#			
	tweets	mentions	direct ref.	indirect ref.
#Yankees	86	228	153	108
#Obama	92	227	167	87
#Ebola	97	241	151	156
#Nobel	94	287	228	124
#Islam	92	219	151	95
Mean per tweet		2.61	1.84	1.24

different annotators as follows. In a tweet, given one annotation in which entity e_1 is referred to by mention m_1 from annotator 1 and another annotation in which entity e_2 is referred to by mention m_2 from annotator 2, if m_1 and m_2 overlap at least one word in the tweet and e_1 and e_2 are similar in their articles, we made a new annotation such that its mention, which is the union of m_1 and m_2 , refers to both e_1 and e_2 . As shown in Table 3.4, the annotation results showed that there were multiple mentions in a microblog (2.61 per tweet) and the number of indirect references was not small (1.24 per tweet). This result implied that acquiring dependency among multiple referred entities and applying features for indirect references were required.

3.5.2 Comparative Methods

We used the existing features and model for comparison. We use the following baseline features from previous studies [42], [47], [52].

3. Entity Identification by CRF Model with Adaptive Dependency

Table 3.5: A list of features.

Feature	Description	Input	Output
Keyword	The similarity between the top words K_e in entity e 's article and mention m .	K_e, m	$\text{sim}(K_e, \{m\})$
Occurrence frequency	The frequency that the name n_e of e and m co-occur in documents.	n_e, m	$\log r(n_e, m)$
Topic-specific translation	The similarity between the context C_e of e and the translated content T_b of microblog b .	C_e, T_b	$\text{sim}(C_e, T_b)$
Entity-known-as pattern	The frequency that n_e , pattern p , m co-occur in order in a sentence.	n_e, p, m	$\log l("n_e...p...m")$
Entity's action	The probability that e , as a subject, co-occurs with verb v_s in a sentence.	e, v_s	$p(e v_s)$
Mention entity's name	The probability that m refers to the name n_e in Wikipedia.	n_e, m	$p(n_e m)$
Context similarity	The similarity between the context C_e and the content W_b of b .	C_e, W_b	$\text{sim}(C_e, W_b)$
Entities' correlation	The similarity between the set of entities V_e linking with e and that set V_{e_2} of entity e_2 .	V_e, V_{e_2}	$\frac{ V_e \cap V_{e_2} }{ V_e \cup V_{e_2} }$
User interest	The similarity between the name n_e and the past microblogs B_u of writer u .	n_e, B_u	$\text{sim}(\{n_e\}, B_u)$

Mention entity name: the probability that mention m refers to the name n_e of entity e by counting the number of anchor-link $a(m, n_e)$ where $a(m, n_e)$ links to the article of e by the text m in an article in Wikipedia, i.e., $p(n_e|m) = \frac{\#a(m, n_e)}{\sum_{e' \in \Gamma} \#a(m, n_{e'})}$ where $\#a(m, n_e)$ is the number of $a(m, n_e)$ in Wikipedia and Γ is a set of entities linked by the text m .

Context similarity: the similarity between the context C_e of e and the content W_b of microblog b , i.e., $\text{sim}(C_e, W_b)$. We also measured the context of e and neighboring entity e_2 in b using $\text{sim}(C_{e_2}, W_b)$.

Entities' correlation: the correlation between two referred entities e and e_2 in b by measuring the similarity between the set V_e of e and that V_{e_2} of e_2 , which is $\frac{|V_e \cap V_{e_2}|}{|V_e \cup V_{e_2}|}$, where V_e is a set of entities, for each of which there exists an anchor-link linking to e in its article or linked by in e 's article.

User interest: to evaluate user interest by computing the possibility that the name n_e of e appears on past microblogs B_u of writer u of b , i.e., $\text{sim}(\{n_e\}, B_u)$.

Baseline model: previous studies [37], [47] ranked candidate entities on a single mention basis as follows. For each mention $m_i \in M$ in a microblog, to select the best candidate entity

3. Entity Identification by CRF Model with Adaptive Dependency

Table 3.6: Parameters used in experiments.

Parameter	Value
# of top candidate entities	1000
# of keywords K_e	0.1k
# of top-k words P_b	0.1k
# of top-k words I_b	1k
# of top-k words Δ	50
# of patterns	10
# of past microblogs B_u per writer	0.1k
Size of words-window for context of entities	0.1k
Threshold of semantic similarity θ	0.7, range [0,1]
ρ for threshold of connection's importance	0.001

$$e' = \operatorname{argmax}_{e_i \in D_{m_i}} p(e_i | e_1, \dots, e_{i-1}, e_{i+1}, \dots, e_n, x), \quad (3.16)$$

where D_{m_i} is a set of candidate entities of m_i and $e_1, \dots, e_{i-1}, e_{i+1}, \dots, e_n$ are currently predicted entities of other mentions. For a fair comparison with our results, the predicted entity of each mention was determined by repeating the above ranking until these predicted entities were not changed or reached 10,000 rounds.

Evaluation metrics. The performance was measured using the mean reciprocal rank (MRR),

$$MRR = \frac{1}{|T|} \sum_1^{|T|} 1/\operatorname{rank}_i, \quad (3.17)$$

where T is a set of tests and rank_i is the ranked position of the ground-truth entity compared with other candidates at a single mention of a microblog by score at test i . Note that if the set of extracted candidate entities for a mention did not contain the ground-truth one, the corresponding $1/\operatorname{rank}_i$ was set as zero. We list the parameters used in experiments in Table 3.6 and we ran 4-fold cross-validation on the experimental dataset for the experimental results.

3. Entity Identification by CRF Model with Adaptive Dependency

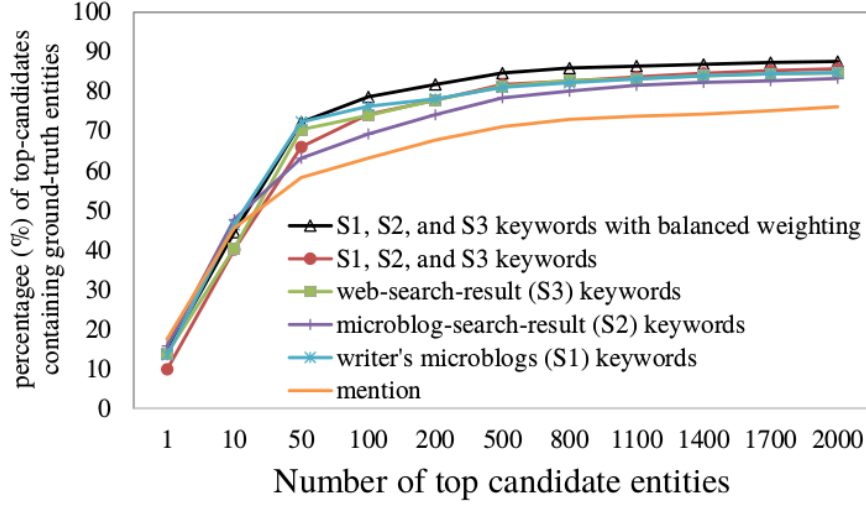


Figure 3.2: Performance of candidate entity extraction.

3.6 Experimental Evaluation

3.6.1 Performance of Candidate Entity Extraction

We evaluated the candidate entity extraction by examining whether ground-truth entities are included in generated candidates. Figure 3.2 shows that with the search query Q of balanced weighting words from different domains, we could boost the performance (2.76% gain when using top-1000 candidates) compared with Q of equally weighing words. In addition, the experimental result showed that extracting Q from different domains (i.e., writer's or users' microblogs or Web) was effective to search for correct entities (3.27%, 4.87%, 3.20%, and 12.65% gain compared with S_1 , S_2 , S_3 , and mention m , respectively, when using top-1000 candidates) compared with a single domain or without searching (i.e., using only mention m). For the experiments of entity identification, we used 1000 candidates for a direct/indirect reference, by which 86.4% of direct/indirect reference cases contained ground-truth entities in the top-1000 candidates.

3. Entity Identification by CRF Model with Adaptive Dependency

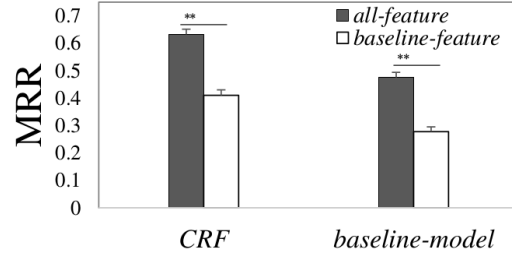


Figure 3.3: Overall performance on models and features (+SEM).

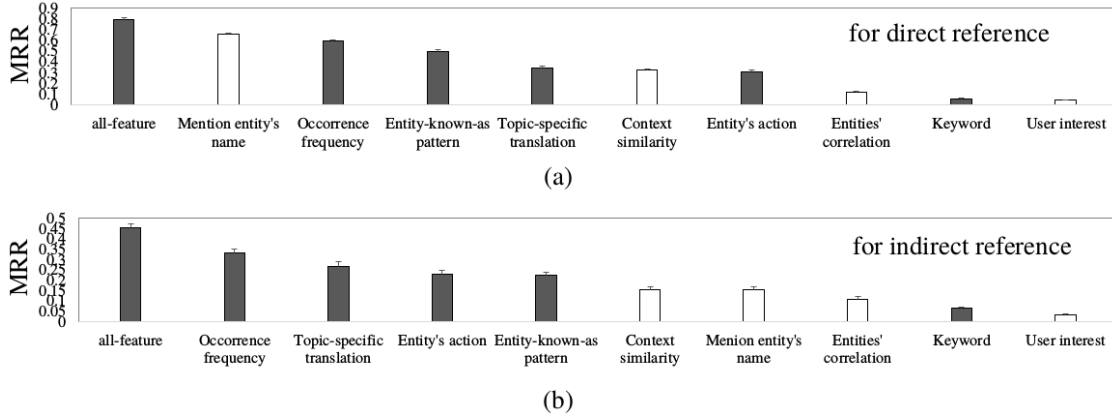


Figure 3.4: The effect of features on (a) direct and (b) indirect references (+SEM).

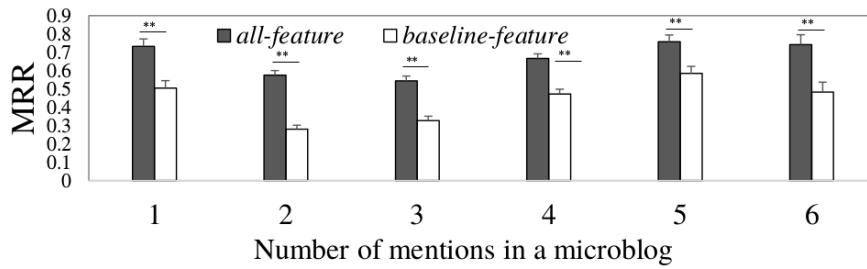


Figure 3.5: Feature comparison (+SEM).

3. Entity Identification by CRF Model with Adaptive Dependency

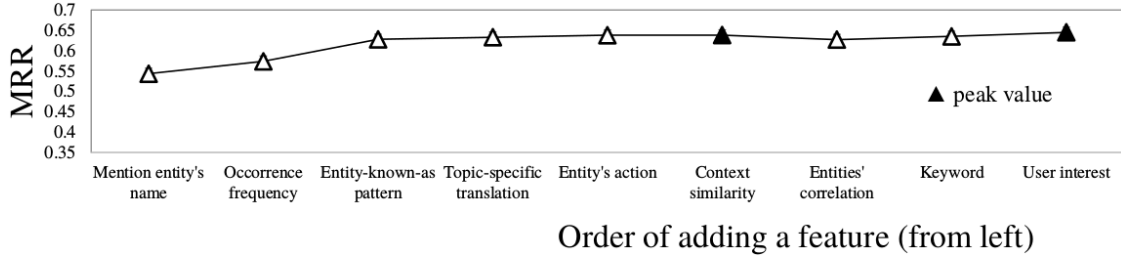


Figure 3.6: Performance of best feature combination (+SEM).

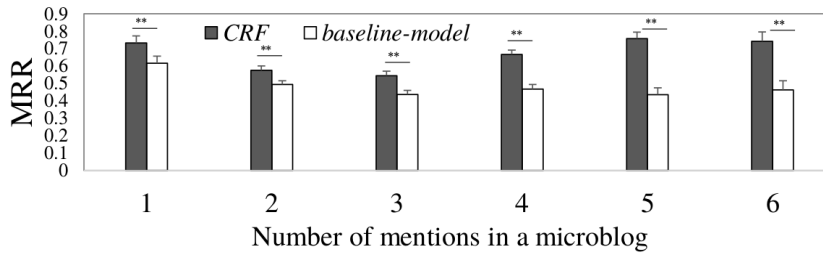


Fig. 7 Model comparison (+SEM).

Figure 3.7: Model comparison (+SEM).

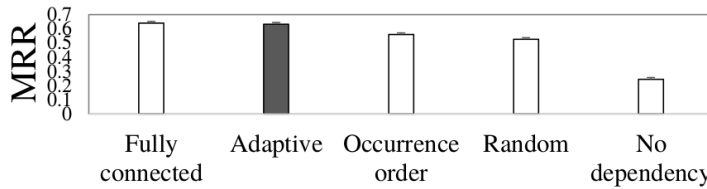


Figure 3.8: Performance of variants of CRF models with varying dependency (+SEM).

3.6.2 Overall Performance

As shown in Figure 3.3, our CRF model using all features achieved the performance of MRR 2.28 times⁵ compared with the baseline model using the baseline features. With CRF model, using all features boosted performance by 1.54 times compared with the baseline features. On using all features, applying CRF model boosted performance by 1.33 times compared with applying the baseline model. The results suggest that not only

⁵ We showed that the experimental results were trustable by applying the t-Test for assessing whether the means between two groups of values were significantly different. We also applied the Bonferroni correction for multiple comparisons. Note that * and ** denote $p < .05$ and $p < .01$, respectively.

3. Entity Identification by CRF Model with Adaptive Dependency

Table 3.7: Case study. Note that ground-truth entities are underlined.

Mention	Candidate entities	Difficulty
“our Suzuki”	“ <u>Jacoby Ellsbury</u> ”, “ <u>Ichiro Suzuki</u> ”, “Suzuki Kei”	Metaphor-like reference
“the prospect”	“ <u>Luis Severino</u> ”, “Karl-Anthony Towns”	Abstract term
“pitcher of Yankees”	“ <u>Masahiro Tanaka</u> ”, “New York Yankees”, “Derek Jeter”	Ambiguous between entity’s name and attribute

our CRF model with adaptive dependency but also integrating existing features with our proposed features for direct and indirect references were important for the entity identification problem. We investigated the effect of ours/the baseline features and CRF/the baseline model in detail.

3.6.3 Feature Comparison

We showed the effect of features on direct/indirect references in Figure 3.4 (a)-(b). Our proposed features performed well on both direct/indirect references, cf. three and four of the top-five single features in direct and indirect references, respectively, were ours. For indirect references, multiple features were more effective than using a single feature as shown in Figure 3.4 (b). The reason might be that the referred entities induced from indirect references were of different types and from different topics; thus we required more features to evaluate them. For direct references, we observed that the existing feature of mentioning entity name was important. Furthermore, we compared the effect of all features with that of the baseline features as shown in Figure 3.5. All features including ours were robust when there were multiple mentions in a microblog. The reason might be that some of our features could obtain quality measurement for multiple referred entities and mentions together, e.g., topic-based semantics and entity-known-as pattern. To identify which features are important, we sequentially added one important feature from currently unselected features into our CRF model as shown in Figure 3.6. We found that for enhancing performance, multiple features, which were effective for direct/indirect references, were required collectively, cf. requiring the top-six features to

3. Entity Identification by CRF Model with Adaptive Dependency

reach the first peak MRR including mention entity name and our entity-known-as pattern and topic-based semantics. Model comparison: as shown in Figure 3.7, we compared our CRF model with the baseline model. The result showed that our CRF model was effective when there were multiple mentions, which suggest that our CRF model with adaptive dependency could acquire proper dependency among multiple entities for correctly inferring multiple entities together, compared with the baseline model.

3.6.4 Model Performance

As shown in Figure 3.7, we compared our CRF model with the baseline model. The result showed that our CRF model was effective when there were multiple mentions, which suggest that our CRF model with adaptive dependency could acquire proper dependency among multiple entities for correctly inferring multiple entities together, compared with the baseline model. We further compared the CRF model of our proposed adaptive dependency with other dependency including connecting each pair of referred entities (fully connected), according to the occurrence order of mentions in the microblog (occurrence order), randomly selecting dependency, and no dependency as shown in Figure 3.8. The results suggest that our CRF model with adaptive dependency was slightly worse than that with the fully connected dependency, cf. the MRR difference was 0.0083; however, a CRF model with the fully connected dependency induced the exponential time-complexity for predicting entities. The dependency of occurrence order provided quality results, which implied that people write the related content together in a microblog and this characteristic is useful to model dependency when we have little information about the dependency.

3.6.5 Case Study

We investigated difficult cases and listed possible difficulty in Table 3.7. One case was a metaphor-like reference, where a mention was obviously the name of a certain entity but was used to refer to another entity. The second case was that a mention was abstract words and thus it could generate candidate entities from a variety of topics. The last case was an ambiguous condition between entity name and attribute. E.g., in the mention “pitcher of Yankees”, “pitcher” is a normal noun and it could target candidate entities of indirect references; however, candidates targeted by “Yankees”, which is a proper noun, may come from direct references.

3.7 Discussion

We listed the findings from the experimental results as follows.

- When identifying multiple indirectly or directly referred entities in microblogs, applying multiple types of features, including common corpus-based features (e.g., keywords and co-occurrence) and specific features (e.g., topic-based semantics), was effective.
- Our adaptive CRF models were effective to model dependency among multiple referred entities. It was also effective when there were more mentions of a microblog.
- Corpus-based features, such as co-occurrence and mention entity name, were effective for indirect or direct references.
- Specific features, such as topic-based semantics or patterns of phrases, were effective to identify certain types of indirect references.

We also discussed the limitation of this research as follows.

- Since the mentions in a microblog were given in the problem setting and we did not detect possible entity mentions in a microblog, we might lack to identify other possible entities.
- We did not apply approaches to measure the correlation of terms in a corpus, such as word2vec models.

CHAPTER 4

4. EVENT IDENTIFICATION FOR IMPLICIT AND EXPLICIT REFERENCES

4.1 Introduction

People write microblogs to share their opinions about information in the real world and refer to events, such as news, in various ways. Event identification, a problem of identifying events referred to by documents, has been addressed extensively in the literature [1-3]. Most related studies addressed the problem of identifying events referred to by event contents, such as subjects and circumstances in events. However, there are many events referred to by microblogs without using event contents, which are often difficult to identify, especially for short text such as microblogs. E.g., given a microblog “United Kingdom’s decision made people unbelievable!”, it is hard to know which event made people unbelievable if one does not know the fact that “United Kingdom’s withdrawal from the European Union was far away from people’s expectations.”. In this research, we address the event identification problem with special attention to implicit reference cases in which events are not referred to by event contents. Especially, if people refer to events by using their opinions or feelings, it is difficult to identify referred events because opinions/feelings are usually irrelevant to event contents.

4. Event Identification for Implicit and Explicit References

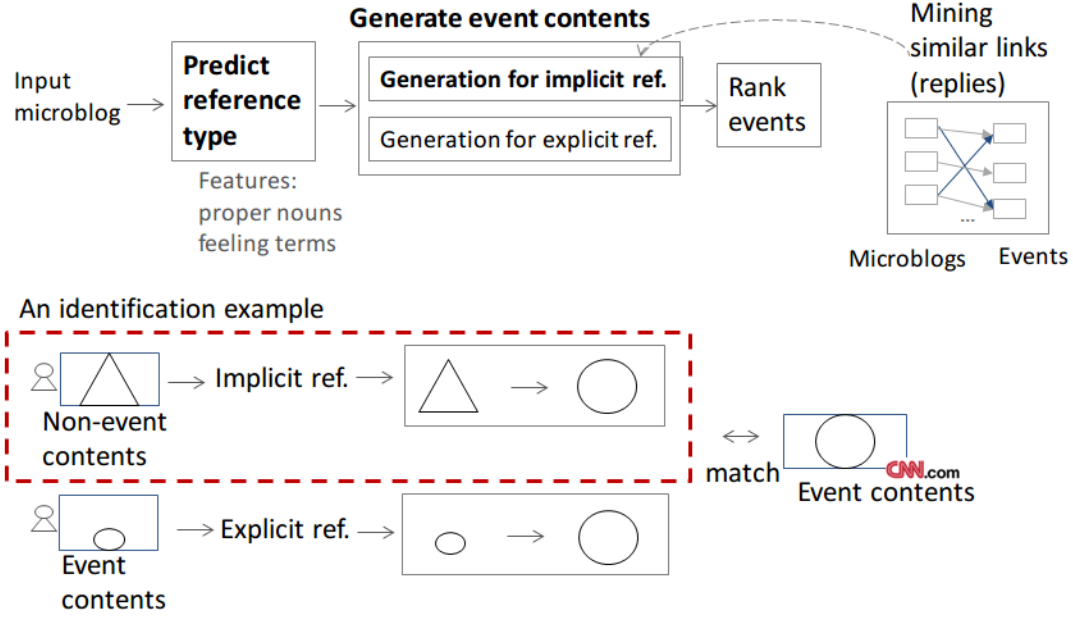


Figure 4.1: We propose reference-type-based event contents generation to identify implicitly or explicitly referred events from microblogs.

We therefore advocate utilizing links from microblogs, on social media platforms, because links are able to connect opinions or feeling. E.g., given a microblog m “It’s a sad news!” linking to an event e “Two injured in gun shooting.”, if a microblog m_2 “It’s a broken-hearted news!” is semantically similar to m , we may infer that microblog m_2 could be used to refer to e . We furthermore find similar links among a set of links from microblogs because similar links contain more clues between microblogs and their referred events. We then propose a clustering-based algorithm for finding similar links.

We also predict reference type of a microblog because referred events can be different according to reference types, E.g., given a microblog “I am a Mysterious fan.” referring to event e “A movie, The Mysterious Men, released.”, if we think that this microblog is an implicit reference because of the feeling of mysterious, we may infer the referred events that make people feel mysterious and will be much different to e . We therefore predict reference type for accurately ranking referred events. Experiments were carried out with 15,996 tweets referring to total 231 events. The experimental results suggested that our method was effective, predicting reference type was essential to identify

4. Event Identification for Implicit and Explicit References

implicitly or explicitly referred events, and finding similar links among microblogs and events was useful to generate contents of implicitly referred events.

We summarize the contributions of this work:

- We address the event identification problem and argue that there is a considerable amount of implicit references in real microblog data.
- To this end, we propose to use reference-type-based event contents generation.
- We demonstrate the effectiveness of our method and that predicting reference type was essential to identify implicitly or explicitly referred events.

The rest of this research is organized as follows. We show the motivation and novelty compared with previous studies in Section 4.1. The event identification problem is presented in Section 4.2. To this end, we propose reference-type-based event contents generation in Section 4.3. We evaluated the proposed methods using real-world microblog data, discussed the experimental results, and presented a case study in Sections 4.5 and 4.6.

4. Event Identification for Implicit and Explicit References

Table 4.1: Event identification problem and related works.

	Problem: event identification for implicit or explicit references	Problem: event identification for explicit references (keyword based)
Ours (Ch 5)	o	o
Ours (Ch 4)	o	o
[4, 8, 28, 29, 114]		o
[3, 11, 20, 24, 111-113, 115]		o
[7, 9, 10, 12, 13, 19, 25-27, 110]		o
[1, 2, 5, 6, 14-18, 23]		o

Table 4.2: Method comparison. We compare on features used for evaluating referred events.

			Features	
	Reference-type-based	Temporal or spatial features	Co-occurrence-based (including co-occurring terms, topic modeling, or word embeddings)	Keyword-based or frequency-based
Ours (Ch 5)	o	o	o	o
Ours (Ch 4)	o		o	o
[4, 8, 28, 29, 114]		o	o	o
[3, 11, 20, 24, 111-113, 115]			o	o
[7, 9, 10, 12, 13, 19, 25-27, 110]		o		o
[1, 2, 5, 6, 14-18, 23]				o

4.1.1 Motivation

Identifying implicitly or explicitly referred events from microblogs

We try to identify the events that are implicitly referred to without event contents. This can be difficult compared with identifying explicitly referred events, in most of previous works [1-20, 23-29, 110-115]. The difficulty is that candidate events can be much different according to reference types. For example, if an adjective term “mysterious” is used to implicitly refer to events, the candidate events can be much more and be in diverse topics. Otherwise, if “mysterious” is used as event contents to refer to an event, the candidates can be limited. Furthermore, to identify implicitly referred events, besides keywords or spatial or temporal features of events, we may consider more factors, such as reference types, writer preferences, and their posting motivations.

4.1.2 Novelty of Our Approach

Reference-type-based event contents generation, with mining similar links among microblogs and events

With the challenging induced by implicit references, we propose novel features and models in Chapter 4 and Chapter 5, respectively. In Chapter 4, we propose to utilize reference type to generate event contents for identifying implicitly or explicitly referred events. This can be useful because the referred events can be much different according to reference types.

In our modeling, we generate contents for implicitly referred events by mining similar links among microblogs and events, and generate contents for explicitly referred events by keywords of events. In contrast, previous works utilized keyword-based or co-occurrence-based modeling (including co-occurring terms, topic modeling, or word embeddings) to identify explicitly referred events as shown in Table 4.2.

4.2 Problem Definition

We introduce several terms to define the problem addressed in this paper as shown in Figure 4.1 . **Microblog**: a microblog is a sequence of words posted by a user on a social media platform. E.g., a tweet posted by a user on Twitter is a kind of microblog. **Event**: an event consists of subjects and the action/status of these subjects. E.g., an event e consists of subject “The Revenant” (a movie) and status “is nominated in Oscar”. We then discuss how an event is referred to by a microblog.

Explicit reference: previous studies [1, 4] generated referred events by using event contents as follows. If microblog m is used to refer to event e and m contains the contents of e , then the reference to e by m is an explicit reference. E.g., m = “The Revenant is mentioned!” refers to event e = “The Revenant is nominated in Oscar.” by using the subject of e . Or, m = “The movie is nominated!” refers to e by the subject’s status. Note that m is used to refer to e if the writer posted m to reply e , where we can check the reply by evidence (e.g., hyperlink or reply of Twitter). However, people can refer to events without using event contents. From real-world microblog data, we found that many people refer to events by posting their opinions, feeling, or related events.

Implicit reference: if microblog m is used to refer to event e and m does not contain the contents of e , then the reference to e by m is an implicit reference as the following cases. E.g., m = “It deserved!” refers to event e by using opinion or m_2 = “Congrats, glad to know this!” refers to e by using feeling. Or, m_3 contains another event “This is directed by Alejandro.”, which is used to refer to e . We therefore define the problem as follows.

Problem Definition (Event identification on microblogs in the presence of implicit or explicit references): given a microblog m from a user u and a set of events E , the problem is to identify the event $e \in E$, which is referred to by m , through implicit or explicit reference.

4. Event Identification for Implicit and Explicit References

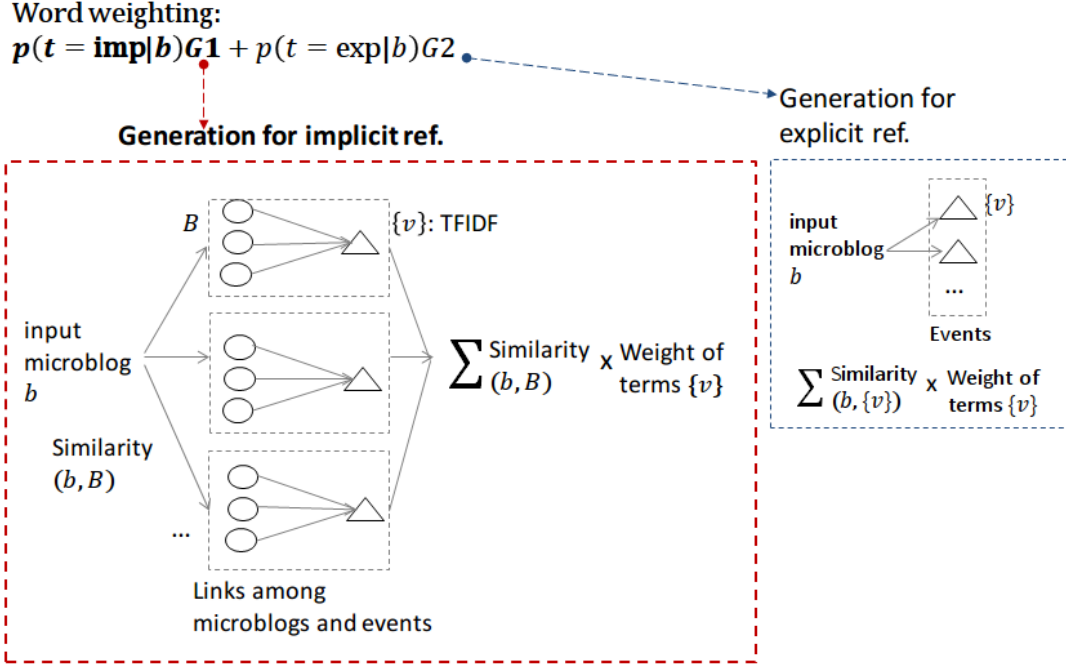


Figure 4.2: Generating event contents for implicit or explicit references.

4.3 Reference-Type-based Event Contents Generation

We propose a method to identify referred events from explicit or implicit references. We especially focus on implicit references and propose a method to generate contents for implicitly or explicitly referred events. We also predict reference type for accurate event identification.

4.3.1 Predicting Reference Type

Given a microblog, we predict reference type of the microblog because the referred event can be different according to the reference type. E.g., given a microblog “I am a Mysterious fan” referring to event e “A movie, The Mysterious Men, released”, if we infer that the microblog is an implicit reference by using mysterious feeling, we may regard the referred event that makes people feel mysterious, which is possibly far from e . Meanwhile, as people usually mention the subjects of events or express the opinions or feelings to the events, we predict the reference types of events by using the types of

4. Event Identification for Implicit and Explicit References

terms. We especially utilize proper nouns or feeling terms and predict the reference type t of microblog m by

$$p(t|m) = \frac{\exp(\mathbf{w}^T \mathbf{f}(t,m))}{\sum_{t'} \exp(\mathbf{w}^T \mathbf{f}(t',m))}, \quad (4.1)$$

where $\mathbf{f}(t,m)$ is a vector of feature values f_1, f_2, f_3 and $\mathbf{w}$ is a vector of weights. $f_1 = n_p$, $f_2 = n_f$, and $f_3 = n_p - n_f$ are the number of proper nouns, the number of feeling terms, and the difference of $n_p - n_f$ in the content of microblog m . We then train $p(t|m)$ by using the objective function of Hinge loss [24], [25],

$$[\max_{\mathbf{y} \in Y} (\Delta(\mathbf{y}, \mathbf{y}^i) + F(\mathbf{y}, \mathbf{x}^i; \mathbf{W})) - F(\mathbf{y}^i, \mathbf{x}^i; \mathbf{W})]_+, \quad (4.2)$$

where $\Delta(\mathbf{y}, \mathbf{y}^i) = \sum_{j=1}^n |1|y_j \neq y_j^i||$ is the hamming distance and $\Delta(\mathbf{y}, \mathbf{y}^i) = 0$ if $\mathbf{y} = \mathbf{y}^i$. By training, we make the compatibility score of the relevant paired entities and mentions larger than that score of other paired entities. Meanwhile, it encourages the score of relevance $F(\mathbf{y}^i, \mathbf{x}^i)$ much larger than other $F(\mathbf{y}, \mathbf{x}^i)$ when the two given entities y and y^i are much different, i.e., $\Delta(\mathbf{y}, \mathbf{y}^i)$ is large. Note that we use the tool [27] for running SSVM training.

4.3.2 Finding Similar Links by Clustering Microblogs and Events Respectively

The difficulty of evaluating an implicit reference between a microblog m and a referred event e is that m does not contain the contents of e , which makes it difficult to identify e . We therefore utilize link because links are able to connect opinions or feeling, which are irrelevant to event contents, to their referred events. A link $r = (s, o)$ on a social media platform is a directed mapping from a source unit of microblog s to a mapped unit o , where o is one of the following two instances. o is another microblog if s links to o . E.g., a user can post s to reply to o in Twitter. Or, o_2 is an outside resource if the content of s contains a link to o_2 . E.g., s contains a hyperlink of a web page o_2 . We argue that if two different microblogs with similar content of text (e.g., opinions) and they link to similar mapped units (e.g., events), we may infer that each microblog can be used to refer to

4. Event Identification for Implicit and Explicit References

another mapped unit. We therefore find similar links for connecting microblogs to their referred events.

It would be difficult to find similar links due to that source units (e.g., opinions) and mapped units (e.g., fact of events) of links come from different domains (types of text). E.g., considering the following equation used for connecting two links $(s, o), (s_2, o_2)$,

$$a * \text{sim}(s, s_2) + b * \text{sim}(o, o_2) \geq \theta, \quad (4.3)$$

where s, s_2 are from domain of source units S , o, o_2 are from domain of mapped units O , $\text{sim}(s, s_2)$ is the cosine similarity between the bag-of-words model of s using tf-idf weighting [30] and that model of s_2 , $0 \leq a, b \leq 1$ are weights, θ is the threshold. It would be hard to compute weights a and b for finding similar links because we do not know the correlation between domain S and domain O . To avoid the above problem, we utilize link instances for building the correlation between domain S and domain O .

Finding similar links by clustering respectively. Given a set of links R , we find similar links by considering similarities of source units (of links), similarities of mapped units, and each link instance. We use clustering for time-efficiently⁶ finding similar source units (or mapped units) of links according to $\text{sim}(s, s')$ (or $\text{sim}(o, o')$). The process of extracting similar links from R is as follows.

1. We extract similar mapped units of links by clustering on each o of $(s, o) \in R$ according to $\text{sim}(o, o')$ using k-means clustering [31] and acquire each cluster as $C_i^o = \{o_{i,1}, o_{i,2}, \dots\}$.
2. For each cluster of mapped units C_i^o , we acquire a set of source units S_i that are linked by each $o \in C_i^o$. I.e., $S_i = \{s | (s, o) \in R, o \in C_i^o\}$.
3. For each linked source units S_i , we then extract similar source units by clustering on each $s \in S_i$ according to $\text{sim}(s, s')$ using *k-means* clustering and acquire each cluster as $C_{i,j}^s = \{s_{i,j,1}, s_{i,j,2}, \dots\}$.
4. According to each cluster of source units $C_{i,j}^s$ and each cluster of mapped units C_i^o , we can extract similar links and generate more linkable source and mapped units of links. I.e.,

⁶ The complexity of computing the similarity between each pair of links r_1 and r_2 , $\forall r_1, r_2 \in R$, is in quadratic, $|R|^2$.

4. Event Identification for Implicit and Explicit References

$$R^* = \{(s, o) | (s, o'), (s, o') \in R, \text{ and } s, s' \in C_{i,j}^S, \text{ and } o, o' \in C_i^O, \forall i, j\} \cup R \quad (4.4)$$

4.3.3 Generating Event Contents for Implicit or Explicit References

Generating event contents for implicit references. Given a microblog m , we find the referred event of m by using a set of links R . However, R^* may not contain the referred event instance. We therefore generate event contents for m as

$$\{(\omega_k, \varphi_k^\alpha)\},$$

where φ_k^α is the weight of word ω_k , as follows.

1. We summarize a set of source units S_i that are linked by each mapped unit o_i in R^* . I.e., $s_i = \{s | (s, o) \in R^*, o = o_i\}$. We then build a bag-of-words model $S_{b,i}$ from the documents of S_i , where $S_{b,i}$ is the summarization of the bag-of-words model of each document in S_i using tf-idf weighting.
2. For each $S_{b,i}$ of source units, we compute the similarity between microblog m and $S_{b,i}$, which is $\text{sim}(m, S_{b,i})$.
3. We sum up the weight of words from each mapped unit o_i using $\text{sim}(m, S_{b,i})$. as follows. Note that we build a bag-of-words model for each o_i as $\{(\omega_k, \varphi_k^{o_i})\}$. Therefore, the weight of each word ω_k and we rank candidate events as

$$\begin{aligned} \varphi_k^\alpha &= \sum_i \text{sim}(m, S_{b,i}) * \varphi_k^{o_i} \\ e' &= \text{argmax}_{e \in E} \text{sim}(e, \{(\omega_k, \varphi_k^\alpha)\}), \end{aligned} \quad (4.5)$$

where E is a set of candidate events referred to by m .

Generating event contents for both references. Since referred events can be different according to reference types, given a microblog m , we compute the probability of reference types of m and generate possible event contents by merging the generation results for implicitly or explicitly referred events, which is

$$\{(\omega_k, \varphi_k)\},$$

as follows.

4. Event Identification for Implicit and Explicit References

1. We build the generation $\{(\omega_k, \varphi_k^\alpha)\}$ for implicit references and build the generation $\{(\omega_k, \varphi_k^\beta)\}$ for explicit references, which is a baseline method of microblog similarity and was shown to be effective for explicit references in experiments.
2. We combine the two generations $\{(\omega_k, \varphi_k^\alpha)\}$ and $\{(\omega_k, \varphi_k^\beta)\}$ with the consideration of probabilities of reference types as follows. We normalize the weight of words in each generation by using max weight and acquire $\{(\omega_k, \varphi_k'^\alpha)\}$ and $\{(\omega_k, \varphi_k'^\beta)\}$. We then sum up the weight of each word ω_k and rank candidate events by

$$\begin{aligned}
 \omega_k &= p(t = \text{"implicit ref"}|m)\varphi_k'^\alpha + \\
 &\quad p(t = \text{"explicit ref"}|m)\varphi_k'^\beta, \\
 e' &= \operatorname{argmax}_{e \in E} \operatorname{sim}(e, \{(\omega_k, \varphi_k)\}),
 \end{aligned} \tag{4.6}$$

where $p(t|m)$ is the probability of reference type t given m .

4. Event Identification for Implicit and Explicit References

Table 4.3: Data summary.

Data	#
Tweets (microblogs)	15,996
Events	231
Implicit (non-event-info.) references	6,810
Explicit (event info.) references	9,186
Writers	10,523
Writers' past microblogs	506,904
Feeling terms	2,628

Table 4.4: Annotation results on 15,996 tweets referring to 231 events.

	Mean per event	Std. error
Tweets referring to	69.25	6.31
Implicit references	29.48	3.65
Explicit references	39.77	4.26

4.4 Experimental Setting

We explained the experimental setting for event identification including the dataset collection, the annotation of microblog data, and comparative methods.

4.4.1 Microblog Annotation

We specifically collected 15,996 tweets (microblogs) from Twitter, each of which referred to one of 231 events, during July 19-August 23, 2016, as follows.

- We collected events by extracting news-like tweets posted by a news-feeder account (e.g., “@cnbrk”) from Twitter. We made sure that each event e was from a news article, which was linked by some news-like tweet m_e via hyperlink. Note that we assume a one-to-many relationship between an event and news articles⁷; thus, an event must correspond to at least one news article.

⁷ <http://edition.cnn.com>

4. Event Identification for Implicit and Explicit References

- We then collected tweets (microblogs) that referred to (linked to) each event e by collecting tweets, which were posted (by Twitter-users) for replying the news-like tweet m_e of e .

We judged the reference type of tweet (microblog) m referring to e by using the following rules.

- The reference type is explicit reference if m contained the information of e .
- The reference type is implicit reference if m did not contain the information of e .

As shown in Table 4.4, the annotation results showed that the number of implicit references was not small (29.48 tweets per event and the ratio of implicit and explicit references was 0.74:1). The result implied that predicting reference type of a microblog and using the predicted type to identify the referred event were required because the possibility of appearance of each type was not small.

4. Event Identification for Implicit and Explicit References

Table 4.5: Parameters used in experiments.

Parameter	Value
# of candidate events	231
# of top-k words (of a bag-of-words model)	100
# of clusters for mapped units	15
# of clusters for the corresponding source units	4
# of iterations of running k-means clustering	1000
ratio of # of top-k links	0.1

4.4.2 Comparative Methods

We used the following baseline method for ranking referred events from previous studies.

Microblog similarity (Ms) [1]: given microblog m , the best referred event is ranked according to the similarity between the content of m and event contents. I.e.,

$$e' = \operatorname{argmax}_{e \in E} \operatorname{sim}(e, m), \quad (4.7)$$

where $\operatorname{sim}(e, m)$ is the cosine similarity between the bag-of-words model trained by e 's content using tf-idf weighting and that model trained by m and E is a set of candidate events.

Best response (Br) [2]: given m , the best e' is ranked according to the top-k links R' from a set of links R . I.e.,

$$e' = \operatorname{argmax}_{e \in E} \operatorname{sim}(e, R'), \quad (4.8)$$

where R' is a bag-of-words model trained by $\{o|(s, o) \text{ is a top-}k \text{ link according to } \operatorname{sim}(m, s), o \text{ is a document}\}$ and we set $k = 0.1|R|$ in experiments.

Writer interest (Wi) [4]: given m , the best e' is ranked according to writer's interest U , where U is a bag-of-words model trained by the profile p of the writer of m and a sequence of past k_2 microblogs $\{m_1, m_2, \dots\}$ of the writer and we set $k_2=50$ in experiments. I.e.,

4. Event Identification for Implicit and Explicit References

$$e' = \operatorname{argmax}_{e \in E} \operatorname{sim}(e, U). \quad (4.9)$$

Evaluation metrics. We measured the performance of event identification by using the mean reciprocal rank (MRR), which is

$$MRR = \frac{1}{|D|} \sum_1^{|D|} 1 / \operatorname{rank}_i, \quad (4.10)$$

where D is a set of tests and rank_i is the ranked position of the ground-truth referred event compared with other candidates for a microblog by score at test i . Note that the ground-truth event is one of candidate referred events.

We ran 4-fold cross validation for the given data and we learned the generation for both implicit and explicit references (denoted as CB) and the generation for implicit reference (denoted as CI) by training data. (3) The cluster size for mapped units and for source units was set 15 and 4, respectively. The feeling terms we used were from this vocabulary list⁸.

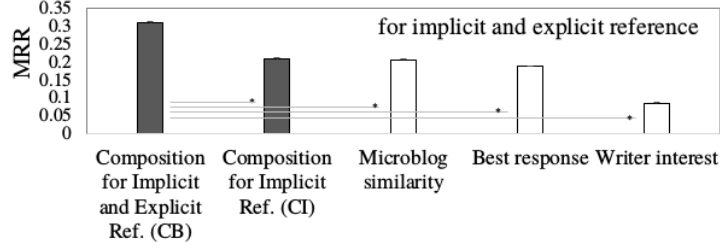
4.5 Experimental Evaluation

4.5.1 Performance on Implicit or Explicit References

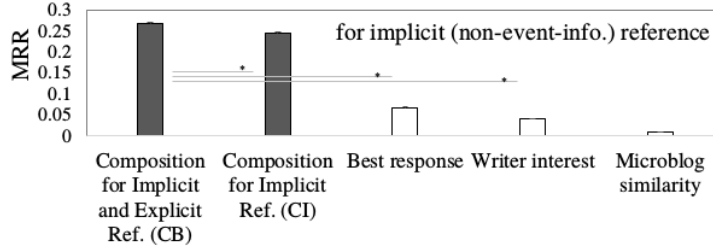
We evaluated each method on event identification as follows. As shown in Figure 4.3 (a), the performance of MRR of our proposed CB was 0.309 and that of the top-one baseline method of microblog similarity was 0.207. I.e., CB achieved the performance of MRR 1.494 times compared with baseline method. The result suggested that generating contents for implicitly or explicitly referred events based on reference types was important. We then investigated the performance on implicit or explicit references in detail.

⁸ <https://www.vocabulary.com/lists/12827>

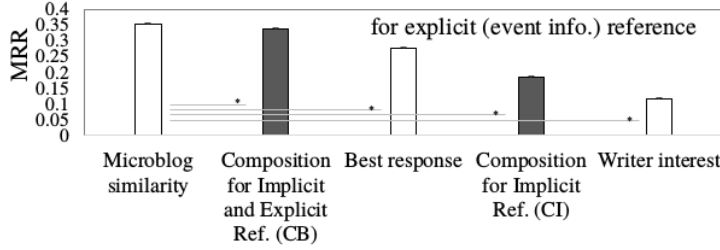
4. Event Identification for Implicit and Explicit References



(a) On implicit and explicit references



(b) On implicit references



(c) On explicit references

Figure 4.3: Comparison of performance on our proposed and baseline methods (+SEM).

Performance on Implicit References

For implicit references, both the proposed CB and only CI were effective than baseline methods as shown in Figure 4.3 (b). E.g., CB achieved the performance of MRR 3.981 times compared with the top-one baseline method of best response. We also observed that the performance of CI was very close to that of CB. The result suggested that generating contents for implicitly referred events was essential and we generated well. We also found that using writer’s past microblogs were not useful for implicit references.

Performance on Explicit References

For explicit references, our CB was slightly worse than the baseline of microblog similarity, cf. the MRR difference was only 0.0135, as shown in Figure 4.3 (c). We also observed that CI did not perform well for explicit references. The reason might be that

4. Event Identification for Implicit and Explicit References

Table 4.6: Performance of reference type prediction.

Reference type	Prediction accuracy
Explicit reference	66.7 %
Implicit reference	58.3 %

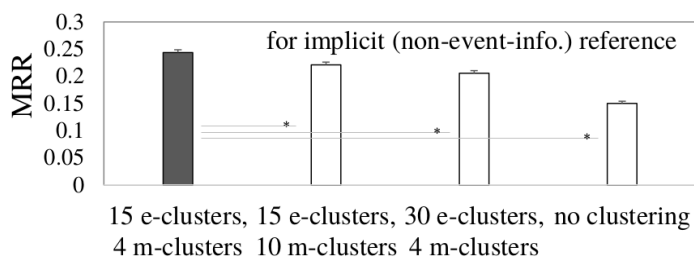


Figure 4.4: Performance of clustering.

CI preferred to link terms that were not event contents. The result suggested that referred events for explicit and implicit references were different. Generating contents for explicitly referred events was needed and could be modeled using keywords of events (e.g., proper nouns of subjects).

4. Event Identification for Implicit and Explicit References

Table 4.7: Case study on implicitly referred events.

Event e	e 's info. "Florida investigating non-travel related case of Zika..."		
Microblogs of implicit reference to e	CB	Method baseline	Difficulty
"@cnnbrk global warming is going to..."	.083	.004 (<i>MRR</i>)	Opinion
"@cnnbrk leave it down there"	.01	.004	Opinion
"@cnnbrk this also outbroke in Brazil"	.2	.004	Related event
"@cnnbrk frightening, but this was..."	.067	.004	Feeling (emotion)

4.5.2 Performance of Reference Type Prediction

We also showed the accuracy of prediction of reference type in Table 4.6. The result showed that our method might infer the reference type of a microblog on some extent. Therefore, we could generate event contents with balanced weight of words for the implicit or explicit references. Note that, for event identification, we used the probability of predicted reference type.

4.5.3 Effect of Clustering

As shown in Figure 4.4, we tuned the performance of the generation for implicit reference (CI) by examining the effect of sizes of clusters of mapped units of links (denoted as e-clusters) and sizes of clusters of source units of links (denoted as m-clusters) when finding similar links. Intuitively, we regarded a cluster of mapped units as a type of events (e.g., "gun abuse") and regarded a cluster of source units as a type of opinions or feeling (e.g., "mysterious"). For implicit references, we observed that modeling CI by setting 15 e-clusters and setting 4 m-clusters for each e-cluster was better than other assignments of more e-clusters or more m-clusters. The result suggested that modeling the generation for implicit references with reasonable parameters might be useful. One reason might be that people reply to a type of events using limited types of responses. E.g., modeling CI with limited types of source units of links (m-clusters was set 4) was better, when e-clusters was 15.

4.5.4 Case Study

We listed some case study in Table 4.7. The performance of identifying for microblogs with related keywords or opinions was better than with emotions. This showed that we should consider more clues to identify for microblogs with emotions.

Statistical Hypothesis Testing

For the overall performance of Figure 3.3 (a), the Analysis of variance (Anova) result ($F(4, 79975)=854.3, p < 2e^{-16}$) showed that at least the effects of two of five methods (*CB*, *CI*, *Ms*, *Br*, and *Wi*, which are denoted by abbreviation) were significantly different. We then ran the Tukey's HSD test, a post-hoc test, and the results (*CB-CI* $p=0.0$, *CB-Ms* $p=0.0$, *CB-Br* $p=0.0$, and *CB-Wi* $p=0.0$) showed that our method *CB* was significantly different with other methods. For the performance on implicit references of Figure 3.3 (b), the Anova result ($F(4, 34045)=1229, p < 2e^{-16}$) showed that at least the effects of two of five methods were significantly different. Then, the results of the Tukey's HSD test (*CB-CI* $p=2.86e^{-5}$, *CB-Ms* $p=0.0889$, *CB-Br* $p=0.0$, and *CB-Wi* $p=0.0$) showed that our method *CB* was significantly different with other methods except *Ms*. For the performance on explicit references of Figure 3.3 (c), the Anova result ($F(4, 45925)=714.9, p < 2e^{-16}$) showed that at least the effects of two of five methods were significantly different. Then, the results of the Tukey's HSD test (*CB-CI* $p=0.0$, *CB-Ms* $p=0.0$, *CB-Br* $p=0.0$, and *CB-Wi* $p=0.0$) showed that our method *CB* was significantly different with other methods. For the performance of Figure 3.4, the Anova result ($F(3, 27236)=73.31, p < 2e^{-16}$) showed that at least the effects of two of four clustering settings, (15 e-clusters, 4 m-clusters), (15,10), (30,4), and no clustering, were significantly different. Then, the results of the Tukey's HSD test ((15, 4)-(15, 10) $p=0.0033$, (15, 4)-(30, 4) $p=0.0$, and (15, 4)-no $p=0.0$) showed that our used setting (15, 4) was significantly different with other settings.

4.6 Discussion

We listed the findings as follows.

- Predicting reference type was essential to identify implicitly or explicitly referred events. The reason might be that referred events were much different according to reference types. In our experiments, the performance of reference-type-based identification was better than that without predicting reference type.
- Mining similar links between microblogs and events was effective for certain types of implicit references. In our experiments, the performance of identifying the types of opinions was better than that of emotion types. Since the identification of emotion types can still be improved, we need to consider more effective modeling or features.

We discussed the limitations of this research as follows:

- We did not detect which part of an event is referred. People may intend to refer to partial contents of an event.
- We did not apply approaches to measure the correlation of terms in a corpus, such as word2vec models.

CHAPTER 5

5. EVENT IDENTIFICATION BY DEEP TRANSFER LEARNING WITH DUAL TRANSFORMATIONS

5.1 Introduction

Event identification, a problem of identifying the referred events from microblogs, is especially difficult when the microblogs do not contain any literal contents of events (e.g., keywords). In Chapter 4, we proposed a method of reference-type-based event contents generation. However, the emotion cases of implicit references are still difficult to identify. The difficulty is that the microblog contents and their referred events in implicit reference cases can be much different and in diverse topics, compared with those in explicit reference cases.

We therefore propose a novel approach that finds new representations for microblogs and events, such that the microblogs and events are easier to be matched in the new representation space. To this end, we propose a deep transfer leaning framework with the following components. (1) Dual transformations (DT): we propose to transform for microblogs and events respectively, since not only microblogs but also events can be

5. Event Identification by Deep Transfer Learning with Dual Transformations

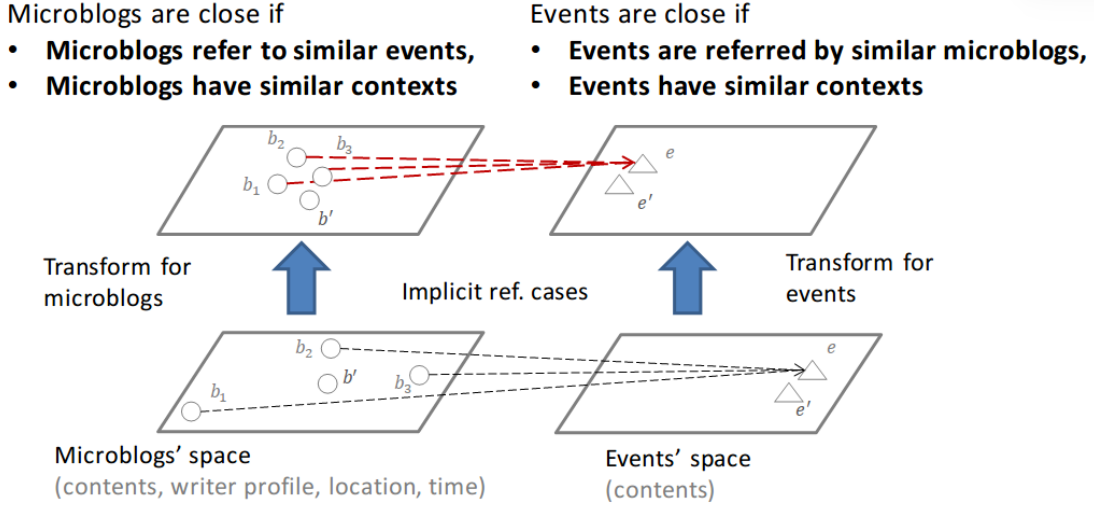


Figure 5.1: We find meaningful similarity between microblogs and implicitly referred events by dual transformations, which are useful for event identification. Note that microblogs and events before transformations are in below, and transformed microblogs and events and their new representation space are in above.

complicated and involve in diverse factors (e.g., opinions/feelings in microblogs vs. knowledge in news events). Note that most of related models did not apply DT and can be seen as using single-direction transformations. (2) All factors can be clues for transformations: to learn effective representations for microblogs and events, we advocate each feature of microblogs and events can be useful. We use the textual contents, writer profiles, posted locations, and time for microblogs, and use the textual contents for events. Note that we expect to use more factors. (3) To identify implicitly referred events, we apply DT with competitive modeling. We apply DT with deep-learning-based models. As predicting reference type was effective, we propose reference-type-based DT. In the modeling, we use a DT model for implicitly referred events and another DT model for explicitly referred events. We summarize the contributions of this work:

- We propose a deep transfer learning framework with dual transformations (DT) to identify implicitly or explicitly referred events.
- To identify implicitly referred events, we propose reference-type-based DT and apply DT with dense modeling that contains more layers of neural networks.

5. Event Identification by Deep Transfer Learning with Dual Transformations

- We demonstrated the effectiveness of our models in extensive experiments for implicit references and the target of unseen events.

The rest of this research is organized as follows. We show the motivation and novelty compared with previous studies in Section 5.1. The event identification problem is presented in Section 5.2. We propose a deep transfer learning framework with dual transformations in Section 5.3. We evaluated the proposed methods using real-world microblog data for the target of unseen events, discussed the experimental results, and visualized the effectiveness in Sections 5.5 and 5.6.

5. Event Identification by Deep Transfer Learning with Dual Transformations

Table 5.1: Method comparison. We compare on probabilistic modeling for event identification problem.

	Transfer learning (or representation learning) with dual transformations	Probabilistic modeling	
		Deep learning models (including CNN models)	Other probabilistic modeling
Ours (Ch 5)	o	o	
Ours (Ch 4)			o
[111-115]		o	
[1-20, 23-29, 110]			o

5.1.1 Motivation and Novelty of Our Approach

Deep transfer learning by dually transforming microblogs and events respectively

With the challenging induced by implicit references, the microblogs implicitly referring to an event can be much literally different, compared with those microblogs explicitly referring to an event. Similarly, the pairs of microblogs and their implicitly referred events can be much literally different, compared with those pairs of explicit references.

Therefore, we propose to transform microblogs and events respectively, for matching them easier. In our dual transformations (DT), the transformed microblogs can be closer to those microblogs that refer to similar events or contain similar contexts of microblogs. Similarly, the transformed events can be closer to those events that are referred by similar microblogs or contain similar contexts of events. We expect DT can be effective for implicit references. Previous works [111-115] have utilized deep learning models including CNN model, but the idea of DT are not yet evaluated in the literature [1-20, 23-29, 110-115].

5. Event Identification by Deep Transfer Learning with Dual Transformations

Table 5.2: Transformation approaches used in related problems.

Transformation approaches					
	Dual transformations from high-dimension side to low-dimension side, and vice versa	Transformations to latent dimensions (including the dimension of seen class proportions)	Single transformation from high-dimension side to low-dimension side	DL models	Link-based (including hierarchy-based, attribute-based, knowledge graph)
Ours (Ch 5)	o		o	o	
[87]			o	o	o
[89, 111-115]			o	o	
[80, 81, 86, 88]			o		
[84, 85]		o			
[82, 83]					o

Dual transformations might be a novel approach compared with single-direction transformations

We furthermore compare on transformation approaches used in identification or classification problems as shown in Table 5.2. In the problems [80, 81, 86-89, 111-115], single-direction transformations from high-dimension side (e.g., image) to low-dimension side (e.g., label) had been proven to be effective. Competitive works also utilized deep learning models [87, 89, 111-115] or link structures or attributes from knowledge database or graphs [82, 83, 87]. However, dual transformations are not yet evaluated and can be useful for multiple cases in implicit references. In our problem, we dually transform for both sides (microblogs and events) because both of microblogs and events are complicated and relates to different factors (e.g., opinions in microblogs vs. knowledge in news events).

5.2 Problem Definition

We introduce several terms to define the problem addressed in this paper as shown in Figure 4.1 . **Microblog**: a microblog is a sequence of words posted by a user on a social media platform. E.g., a tweet posted by a user on Twitter is a kind of microblog. **Event**: an event consists of subjects and the action/status of these subjects. E.g., an event e consists of subject “The Revenant” (a movie) and status “is nominated in Oscar”.

Problem Definition (Event identification on microblogs in the presence of implicit or explicit references): given a microblog m from a user u and a set of events E , the problem is to identify the event $e \in E$, which is referred to by m , through implicit or explicit reference.

5. Event Identification by Deep Transfer Learning with Dual Transformations

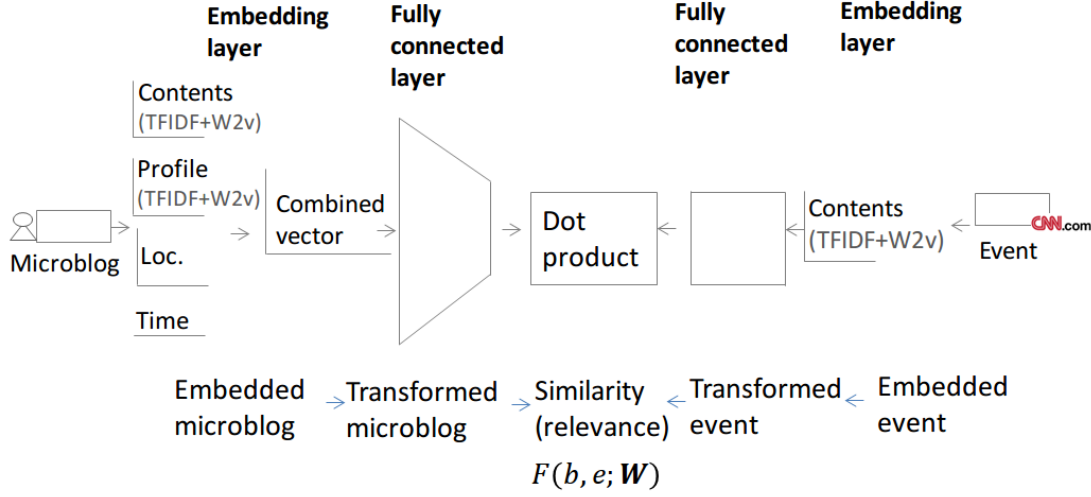


Figure 5.2: A basic model of deep transfer learning with dual transformations. The inputs are a microblog (left) and an event (right), and the output is their relevance of dot product after dual transformations.

5.3 Deep Transfer Learning with Dual Transformations

We detail the framework of deep transfer learning with dual transformations (DT). For effective transformations, we utilize all factors related to microblogs and events and represent as the vectors including textual contents of microblogs and events, writer's profiles, posted locations and time. The vectors of microblogs and events are transformed respectively with different transform functions, finally into a space of the same dimension, and their relevance can be estimated by the inner product of their transformed ones. To identify implicitly referred events, we apply DT with dense modeling or with CNN-based models. Note that we can estimate relevant microblogs for unseen events, which do not exist in training data, because this framework estimates relevance based on transformed vectors. Once we learn the matching for those transformed ones, which are similar to transformed ones of unseen events, we can estimate relevance for unseen events.

5.3.1 Embedding Vectors for Microblogs and Events Respectively

Microblogs are characterized by their textual contents, writer's profiles⁹, and posted locations and time. In our experiments, each microblog b was represented as a vector $\mathbf{b} = (\mathbf{b}_c, \mathbf{b}_u, \mathbf{b}_l, \mathbf{b}_z)$. The content vector $\mathbf{b}_c$ was constructed as the weighted average of the word embeddings of the terms in the content

$$\mathbf{b}_c = \frac{1}{z} \sum_v TF-IDF(v, b_c) WE_c(v), \quad (5.1)$$

where v is each term of the content b_c , $TF-IDF(v, b_c)$ indicates the weight of v in b_c , $WE_c(v)$ indicates the vector of v in the word embeddings space for the contents, and $z = \sum_v TF-IDF(v, b_c)$ is used for weighted average. The writer's profile vector $\mathbf{b}_u$ was constructed in the same way:

$$\mathbf{b}_u = \frac{1}{z} \sum_v TF-IDF(v, b_u) WE_u(v), \quad (5.2)$$

where v is each term of the writer's profile b_u and $WE_u(v)$ indicates the vector of v in the word embeddings space for writer's profiles.

The location vector $\mathbf{b}_l$ was a two-dimensional vector indicating the latitude and longitude of the point from which the microblog was posted or the location of the writer's profile. Note that we preprocessed for microblogs that did not have posted locations in experiments. The post time b_z was simply represented by a scalar.

The vector of an event e was constructed in the same way above but used the word embeddings space WE_e for the event contents, which is

$$\mathbf{e} = \frac{1}{z} \sum_v TF-IDF(v, e) WE_e(v). \quad (5.3)$$

Note that we used different models for the word embeddings of microblogs and events, each of which was trained by a set of microblogs and a set of events, respectively, since the same term may have different senses in different media.

⁹ Twitter allows each user to construct a profile describing their background, interests, and other distinguishing attributes.

5.3.2 Dual Transformations for Microblogs and Events Respectively

Specifically, the relevance of a microblog b and an event e is estimated by:

$$F(b, e; \mathbf{W}) = \text{ReLU}(\mathbf{W}_b \mathbf{b} + \boldsymbol{\delta}_b)^T \text{ReLU}(\mathbf{W}_e \mathbf{e} + \boldsymbol{\delta}_e) \quad (5.4)$$

$\mathbf{b} \in R^{n_b}$ and $\mathbf{e} \in R^{n_e}$ are the vector representations of b and e , respectively, which are detailed as the above contents. We transform the representation of microblog b by using a FC layer with the activation function of ReLU, i.e., $\text{ReLU}(\mathbf{W}_b \mathbf{b} + \boldsymbol{\delta}_b) \in R^{n_e}$, where $\mathbf{W}_b \in R^{n_e \times n_b}$ is a trainable matrix and $\boldsymbol{\delta}_b \in R^{n_e}$ is trainable biases. We also transform the representation of event e by using the same way but the FC layer has matrix $\mathbf{W}_e \in R^{n_e \times n_e}$ and biases $\boldsymbol{\delta}_e \in R^{n_e}$.

Pairwise ranking loss

The transformation functions by ReLU are trained by pairwise ranking loss [78]. When an event e is paired with a relevant microblogs b , we aim to make their relevance score $F(b, e)$ bigger than those scores paired with other events. Furthermore, when two events (one is event e and another event e') are dissimilar, we make the difference of their relevance scores to be larger. That is, $F(b, e)$ is expected to be much larger than $F(b, e')$. The pairwise ranking loss:

$$L_f(\mathbf{W}) = \sum_{e' \in E_{tr}} [\Delta(e', e) + F(b, e'; \mathbf{W}) - F(b, e; \mathbf{W})]_+, \quad (5.5)$$

where E_{tr} is a set of training events and $[x]_+ = \max(0, x)$. The difference between e and e' , $\Delta(e', e)$, is 0 if e and e' are identical; otherwise, $\Delta(e', e) = \beta$. In our experiments, β was set to 0.3. Note that we optimize F by stochastic gradient descent with the gradient $\nabla_{\mathbf{W}} L_f(\mathbf{W})$.

We explain how this model could match for unseen microblogs and events. As shown in Figure 5.2, given a pair of unseen microblog b and unseen event e , this model was able to match b and e when their respectively transformed ones $\mathbf{b}^{tr}$ and $\mathbf{e}^{tr}$ had been matched closer during training phase.

5.3.3 Applying DT with Dense Modeling or with CNN Model

To identify implicitly referred events, we apply DT with the dense modeling that contains more layers of neural networks or with CNN-based models [108]. Applying DT with more layers of neural networks can be denoted as

$$F_{dense}(b, e; \mathbf{W}) = (\text{ReLU}^{(n)}(\mathbf{b}))^T \text{ReLU}^{(n)}(\mathbf{e}), \quad (5.6)$$

where $n \in \mathbb{Z}^+$ and $\text{ReLU}^{(2)}(\mathbf{b}) = \text{ReLU}(\text{ReLU}(\mathbf{b}))$ if $n = 2$. Note that embedding vectors $\mathbf{b}$ and $\mathbf{e}$ can be transformed respectively for multiple times by ReLU. Furthermore, applying DT with a CNN model can be denoted as

$$F_{CNN}(b, e; \mathbf{W}) = (\text{CNN}(\mathbf{b}))^T \text{ReLU}^{(n)}(\mathbf{e}), \quad (5.7)$$

where $\text{CNN}(\mathbf{b})$ comprises convolutional layers and pooling layers iteratively for transforming the input embedding vector $\mathbf{b}$.

5.3.4 Reference-Type-based DT

In chapter 4, we showed the effectiveness of predicting reference type in generating event contents for implicitly or explicitly referred events. To increase performance, we therefore propose reference-type-based DT. In this modeling, we use a single DT model for implicitly referred events and another DT model for explicitly referred events, as follows.

$$p(\text{"implicit ref"}|b)F_{imp}(b, e; \mathbf{W}) + p(\text{"explicit ref"}|b)F_{exp}(b, e; \mathbf{W}), \quad (5.8)$$

where $p(\text{"implicit ref"}|b)$ is the probability that microblog b refers to an event by implicit reference.

5. Event Identification by Deep Transfer Learning with Dual Transformations

Table 5.3: Microblog statistics.

Data	#
Tweets	31,174
Tweets referring to events	16,535
Events (news)	108
Writers	17,326
Implicit references	10,809
	(100.08 per event)
Explicit references	5,726
	(53.02 per event)

Table 5.4: Event (news) annotation and text similarities between events.

Category	Unique news: (total #:108)	Consecutive news
POLITICS	27	(4,3,2,2)
CELEBRITY	26	(4,3,2,2)
SOCIETY	18	
WORLD	10	(2)
CRIME	18	(6)
DISASTER	3	
TERROR	6	(3)

Text similarity [0.0, 1.0] between two news	#
[0.0-0.1)	5592
[0.1-0.2)	146
[0.2-0.3)	18
[0.3-0.4)	16
[0.4-0.5)	4
[0.5-0.54)	2
≥ 0.54	0

5.4 Experimental Setting

We explained the experimental setting including the dataset collection and annotation, comparative methods, the dataset setting for evaluation using unseen events, and the tuning of our DT models.

5.4.1 Dataset Collection

For the microblog search experiments, we collected 31,174 tweets (microblogs) from Twitter, among which 16,535 tweets referred to a specific news (event) and the others were randomly selected. The collection procedure is detailed below:

1. News (event) were collected from articles in CNN website¹⁰. We collected news-like tweets posted by the news-feeder accounts (“@cnnbrk”) in Twitter. We ensured that each news-like tweet m_e was unique by its tweet ID and linked to a news e in CNN website.
2. Two human annotators checked the identity of collected news by their contents. As a result, there were total 108 unique news (no duplicate) and some of them were consecutive news, which is detailed below.
3. We collected tweets that replied the news-like tweet m_e with a comment. These tweets were regarded as microblogs referring to news e .

Check identity of events

We checked the identity of news by the following definitions. News: the circumstances about the subject of entities (e.g., people/object), for a specific time or place. Consecutive news: two or more news are considered as consecutive if their circumstances are different but used to describe the same subject. E.g., news 1 “Chris Brown and two other people arrested in Paris on allegations of aggravated rape and drug violations” and news 2 “Singer Chris Brown has been released from police custody and he won’t face any charges at this time” were consecutive news as there are two different circumstances about the same subject (Chris Brown).

Two annotators labeled the category of each news. Within each category, they then checked if each news is unique and if two or more news are consecutive news. Table 5.4 shows the annotation results and “(4,3,2,2)” in POLITICS means there were 4, 3, 2, and 2 news that were consecutive news, respectively. We furthermore counted the similarity of the textual contents between each pair of two news and the results showed that the contents of most news were much different.

¹⁰ <https://edition.cnn.com>

5. Event Identification by Deep Transfer Learning with Dual Transformations

Table 5.5: Preprocess of microblogs' locations.

Microblogs with posted locations	Microblogs with locations of writer's profiles	Microblogs without location data (replaced by random locations)
922 (3.2%)	14065 (49.5%)	13419 (47.2%)

Table 5.3 shows the statistics of the collected tweets. Implicit references are microblogs that refer to an event e without containing any terms in e , whereas explicit references contain terms that are also included in e . Clearly, our dataset contained more implicit references than explicit references. This trend supports our motivation of employing the DNN model for relevance estimation.

Preprocess for microblogs' locations

In real microblog data, such tweets, it often finds that microblogs were posted without attached locations. We preprocessed for the microblogs that did not have posted locations as follows. For a microblog without posted location, (1) we used the location of the writer's profile instead. (2) If (1) did not work, we used a random location instead, which a random one from all existing posted or writers locations.

The consideration of (2) is that since location was used as one factor that measured the similarity between microblogs and events, we must assign one value for the location. Meanwhile, we assign a random value for location to reduce the effect for the model that computes the similarity. In other words, suppose we did not assign the random but one specific value for those microblogs without location, then those microblogs would be considered as the same for their location values, which was not true in real world. Table 5.5 shows the fraction of data sources of microblogs' locations. Since there are more than the half (52.7%) of microblogs that contained either posted locations or locations of writer's profiles, assigning a random location for microblogs without location data might not affect much the similarity computation that needed locations.

5. Event Identification by Deep Transfer Learning with Dual Transformations

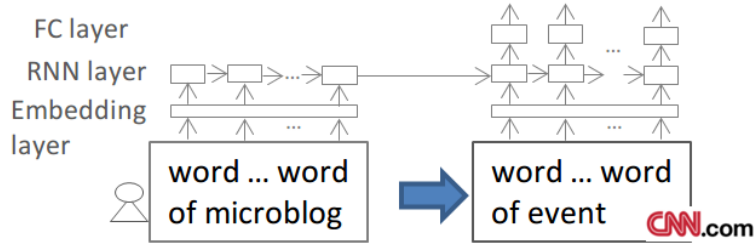


Figure 5.3: A word-sequence mapping from microblog to event, by a sequence-to-sequence model.

5.4.2 Comparative Methods

We compared our method with the following methods from existing studies.

(1) **Deep-learning-based single-direction transformations** (*TRANS*) [87, 89, 111-115]:

This approach denotes that the models, which are used in identification or classification problems, utilize the single-direction transformations techniques, especially for those deep learning based models. We compare on two levels of single-direction transformations. (a) We want to evaluate the effect of single-direction transformations either from microblogs ($TRANS_b$) or from events ($TRANS_e$). In our problem, there can be more and diverse microblogs that refer to a same event, and thus the single-direction transformation from microblogs might be useful to generalize the linking between microblogs and events, compared with the transformations from events. (b) We want to evaluate whether dual transformations still work when applying with competitive models, such as with more layers of neural networks or with CNN-based models.

(2) **Blog-to-event-based matching** (B2E) [102-104]:

This approach finds the relevance between microblogs and events by utilizing the mapping possibility from terms of microblogs into terms of events in sequence. Since events and their associated microblogs have different vocabularies from different domains, this B2E approach utilizes a sequence-to-sequence encoder-decoder model, which is also a deep learning model, as shown in Figure 5.3.

5. Event Identification by Deep Transfer Learning with Dual Transformations

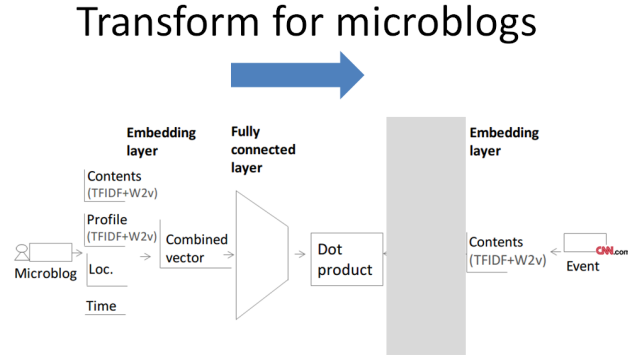


Figure 5.4: A model of single-direction transformations for microblogs.

(3) **Corpus-based matching (CS)** [68, 73, 97-101]:

This approach is an intuitive way to find relevance between microblogs and events based on the correlation of contexts of terms and their frequencies. Meanwhile, the state-of-the-art word embeddings models are applied.

(4) Effect of dual transformations on incomplete features:

We evaluate the effect of dual transformations when applying different features. We evaluate the performance on the word embedding of word2vec models, on the embeddings of one-hot codes (TF-IDF) that provide less correlation among terms, and on the models without using temporal and location features.

5. Event Identification by Deep Transfer Learning with Dual Transformations

Table 5.6: The dataset setting for training and testing. Note that the events used in training and testing were disjoint.

Training phase	Testing phase (Online Searching)
# of seen events $ E_{tr} $: 85	# of unseen events $ E_{test} $: 23 (Note $E_{tr} \cap E_{test} = \emptyset$)
# of microblogs referring to events E_{tr} : 12,875	# of microblogs referring to events E_{test} : 3,660
	# of microblogs without reference: 14,639

5.4.3 Evaluation by Unseen Events

Using the dataset as shown in Table 5.6, in training phase, our and comparative models were trained by 80% of the training data and tuned the hyper parameters by the remaining 20% of the training data. In testing phase, we used the events E_{test} that were not included in the training data, $E_{test} \cap E_{tr} = \emptyset$, which was also called unseen events. Moreover, we included the random microblogs without references for simulating the microblog search problem by a search engine. In testing phase, there was about 1% of all microblogs that referred to a specific event.

5. Event Identification by Deep Transfer Learning with Dual Transformations

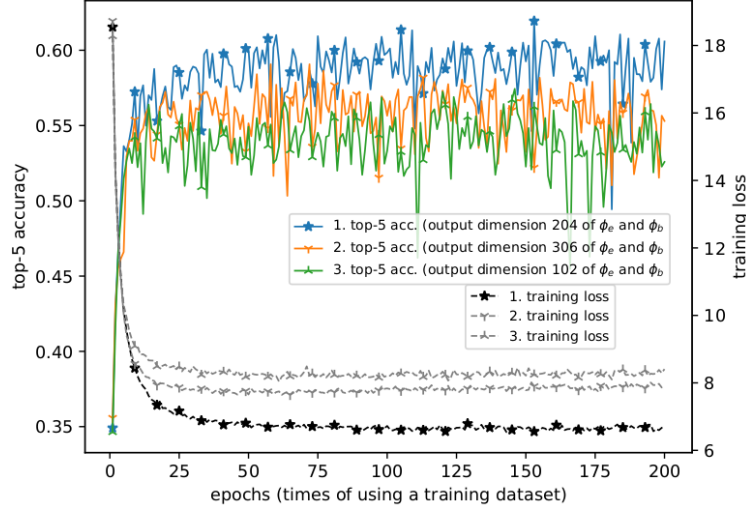


Figure 5.5: Tuning of space dimension after dual transformations. Note that the score is the accuracy of ranking ground-truth events in top-5 positions.

5.4.4 Space Dimension after Dual Transformations

For the size of the DNN model, we tested on which dimension was effective for transforming microblogs and events, respectively, into a space that relevant pairs could be matched easier. In related tasks of classifying images by labels, the dimension after transformations could be the dimension of word embeddings for labels, which can be much smaller than the dimension of CNN-based features for images (e.g., the level of 10^3). We therefore evaluated on a small, equal, and large dimensions compared with the dimension of vector embeddings for events. In the training of the DNN, we found that the dimension of events was effective for the dual transformations. We also decide to use the model's weights at epoch (one time of using the training dataset) 153, because the validation results were not improved with more epochs and the training loss was apparently low at that time.

5.5 Experimental Evaluation

We answered these research questions through the extensive experiments.

- **RQ1.** Can our method effectively identify implicitly or explicitly referred events for given microblogs? Can ours outperform baseline methods such as corpus-based and blog-to-event-based matching? Is it also applicable to unseen events that did not exist in training data?
- **RQ2.** Can transfer learning with dual transformations outperform single-direction transformations, in competitive models? Do dual transformations still work when applying with competitive models, such as with more layers of neural networks or with CNN-based models?

Evaluation metrics

The ranking metrics in experiments is normalized discounted cumulative gain (NDCG):

$$NDCG_k = \frac{1}{z} \sum_{i=1}^k \frac{rel_i}{\log_2(i+1)}, \quad (5.9)$$

where rel_i is one if the ranked item at position i is relevant and z is the optimal value of this ranking.

5. Event Identification by Deep Transfer Learning with Dual Transformations

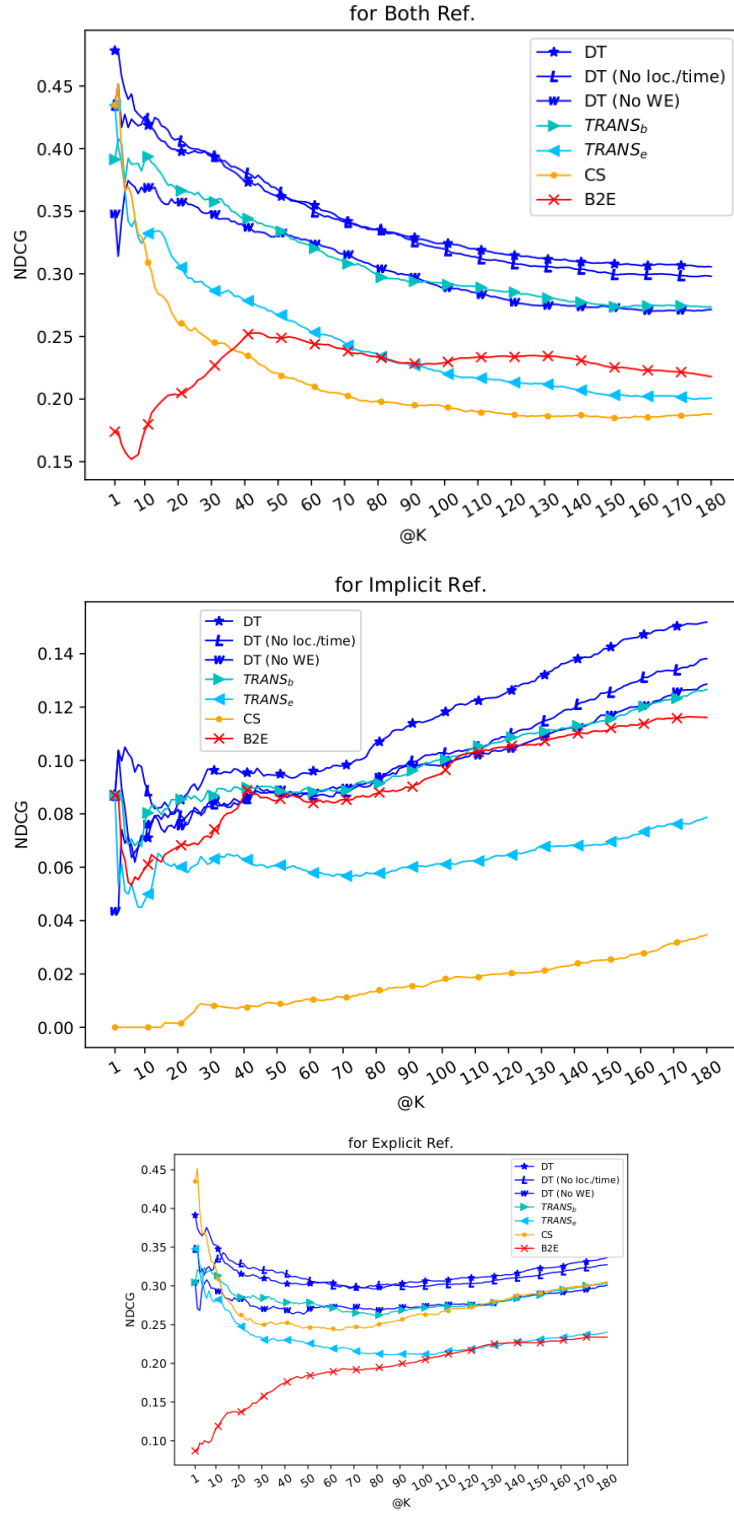


Figure 5.6: Performance of ranking relevant microblogs for events on each method.

5. Event Identification by Deep Transfer Learning with Dual Transformations

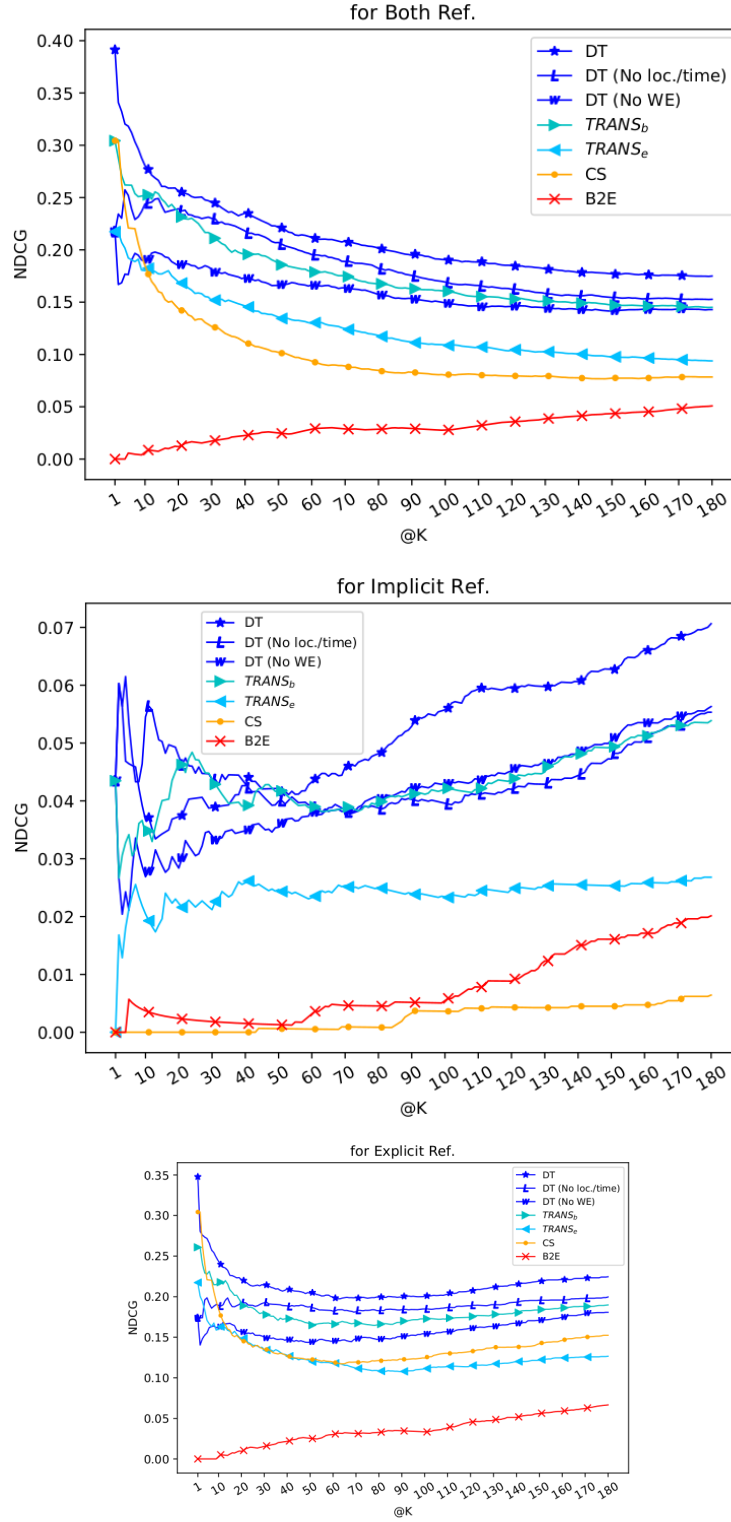


Figure 5.7: Performance of ranking relevant microblogs for events on large-scale microblogs.

5.5.1 Performance of Dual Transformations (DT)

Figure 5.6 showed the performance of ranking microblogs for events, for a set of unseen events. Our dual transformations (DT) outperformed the baseline methods, including corpus-based, blog-to-event-based, and single-direction-transformations based. Single-direction transformations from microblogs ($TRANS_b$) were better than that from events ($TRANS_e$). Blog-to-event (B2E) had quality performance for implicit references but was weak for explicit references, while corpus-based method had the inverse results. Furthermore, when there are more microblogs including those without referring to events, our DT could keep the performance while the performance of blog-to-event decreased much, as shown in Figure 5.7. Moreover, DT without word embeddings (DT (No WE)) was worse than that without location and time (DT (No loc./time)).

The results might imply as follows. (1) The transfer learning with dual transformations was robust. DT outperformed baseline methods in both types of references, even in ranking large-scale microblogs for events. DT could keep performance when applying with incomplete features. However, for modeling DT, diversely underlying features could be useful. The evidence was that DT with word2vec was better than DT without location and time, and followed by DT with TFIDF weighting. Note that TFIDF is one-hot codes that capture less correlation among terms. (2) Dual transformations were better than single-direction transformations, and transformations from microblogs were effective for identification. Note that DT with incomplete features could be competitive with top single-direction transformations in some cases. The reason might be that both of microblogs and events could be complicated and related to multiple factors (e.g., posting behaviors from microblogs vs. knowledge from news events) and thus transforming each of them respectively could be effective for matching them easier. (3) For matching microblogs and events, our mapping between document-to-document level (e.g., one microblog is an embedding vector) was more effective than the mapping between word-to-word level (e.g., B2E). The reason might be that events were hard to match by word-to-word of microblogs, which is usually grammar free and lack sequence of words. We answered **RQ1** that our model could identify relevant microblogs for both types of references and outperformed each baseline method, for unseen events.

5. Event Identification by Deep Transfer Learning with Dual Transformations

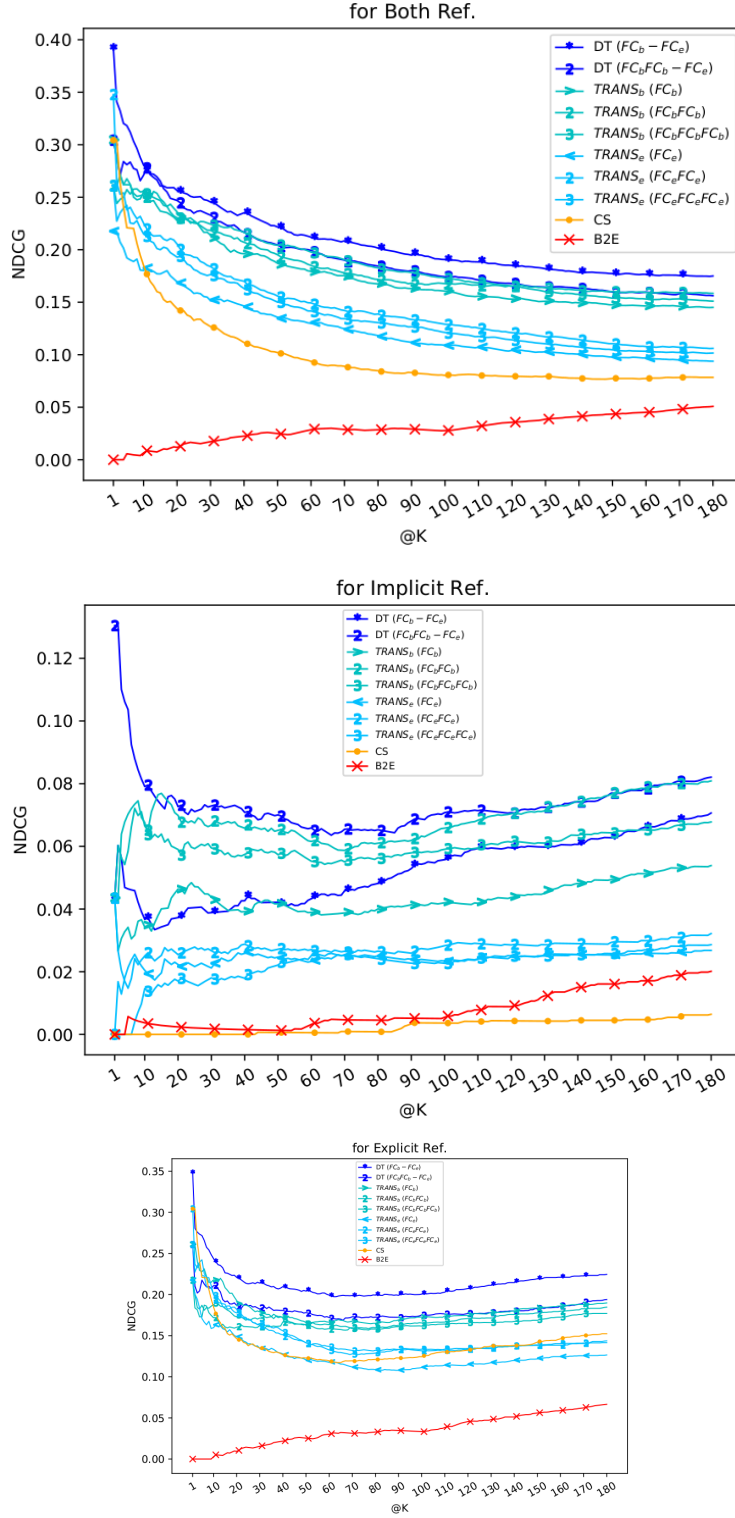


Figure 5.8: Performance of dual transformations and single-direction transformations with dense modeling.

5. Event Identification by Deep Transfer Learning with Dual Transformations

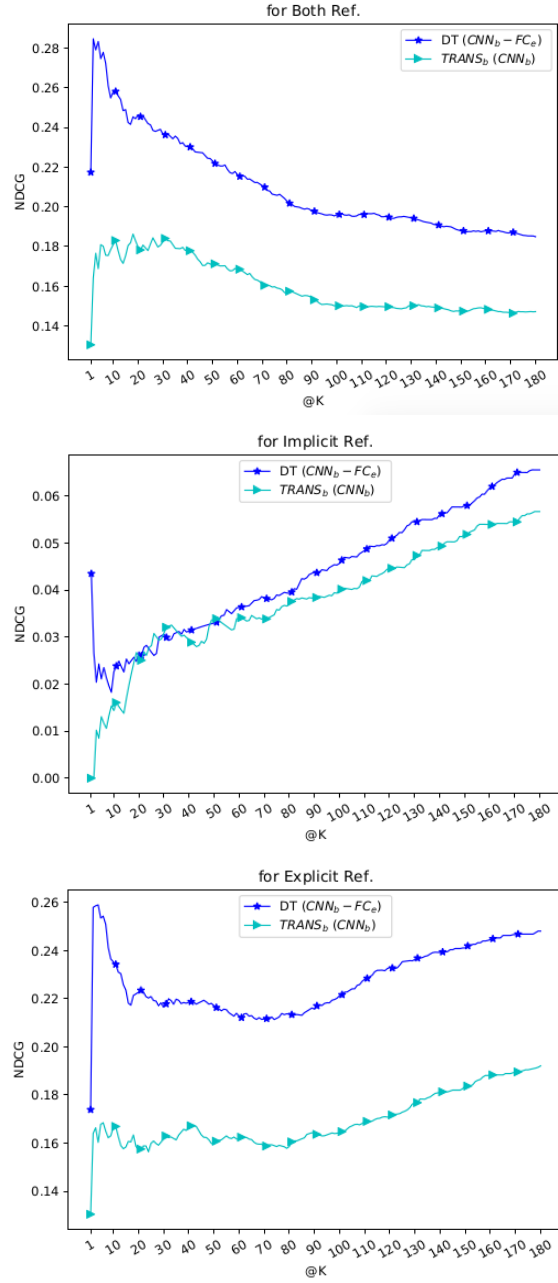


Figure 5.9: Performance of dual transformations and single-direction transformations with CNN model.

5.5.2 Performance of DT with Dense Modeling or with CNN Model

Since common deep learning models can have more layers of neural networks or modeled with competitive models, such as CNN model, we furthermore evaluate whether dual transformations still work when applying on more layers of neural networks or with CNN-based models.

Figure 5.8 showed that when applying more layers, dual transformations (DT ($FC_bFC_b - FC_e$)) and single transformations from either microblogs ($TRANS_b(FC_bFC_b)$) or events ($TRANS_e(FC_eFC_e)$) increased the performance on the implicit references cases, but DT and $TRANS_b$ decreased their performance on explicit references. Moreover, $TRANS_b$ could achieve same level of performance compared with our DT on implicit references. We also observed that adding more layers, such as ($TRANS_b(FC_bFC_bFC_b)$), might not increase as expected. Figure 5.9 showed that when applying with CNN-based models, the DT could increase the performance of the CNN model on explicit reference cases. Note that the CNN model consisted of one layer of CNN and one layer of Max Pooling.

The results might imply as follows. (1) Dual transformations were still better than single-direction transformations from microblogs, in a small degree. (2) Dense modeling, which increased the layers of neural networks, was effective for either dual or single-direction transformations on implicit references, but was not useful on explicit references. (3) On the other side, for CNN-based models, applying DT could increase the performance on explicit references. (4) Although modeling DT by dense layers or with competitive modes, such as CNN model, could be useful, one best model was the DT of original version and this implies that we should carefully design dual transformations based on the problems. We answered **RQ2** that DT was more effective than single-direction transformations. Meanwhile, DT with dense modeling was effective on implicit references and DT with CNN-based models was effective on explicit references.

5. Event Identification by Deep Transfer Learning with Dual Transformations

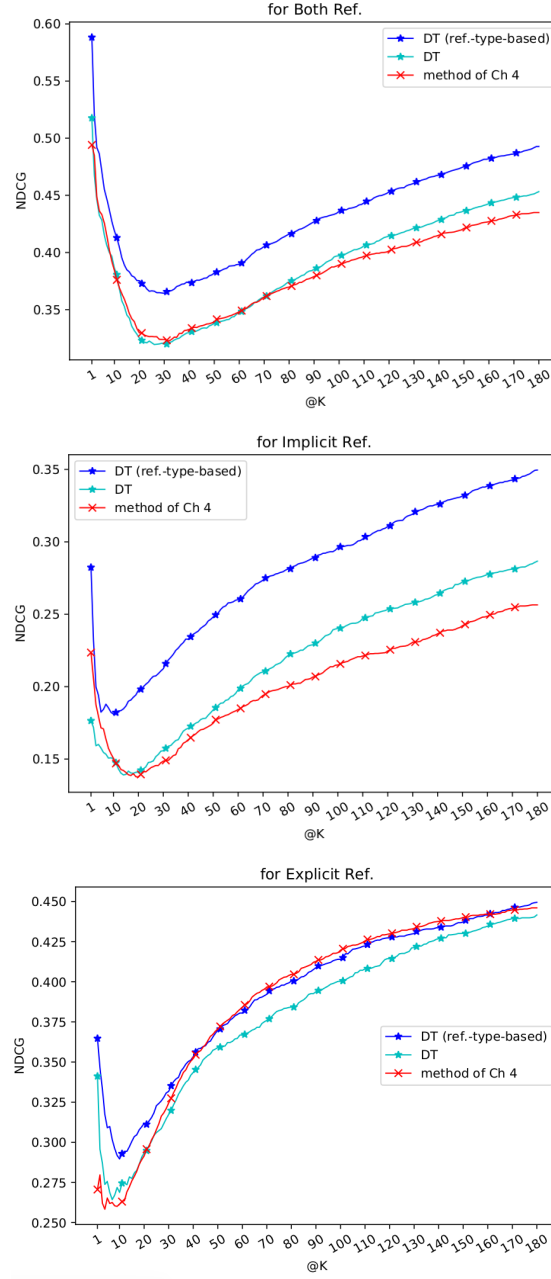


Figure 5.10: Performance of reference-type-based DT and the method of Chapter 4.

5.5.3 Performance of Reference-Type-based DT

In Figure 5.10, we compared DT with the method of Chapter 4 and the effect of predicting reference type. The results were evaluated on seen events since the method of Chapter 4 cannot be applied to unseen events. In reference-type-based DT, we used two DT models with dense modeling (DT ($FC_b FC_b - FC_e$)) for implicit or explicit references respectively. The results showed that DT was more effective in implicit references than the method of Chapter 4, and predicting reference type was effective for identifying implicitly or explicitly referred events, on the modeling of DT.

5. Event Identification by Deep Transfer Learning with Dual Transformations

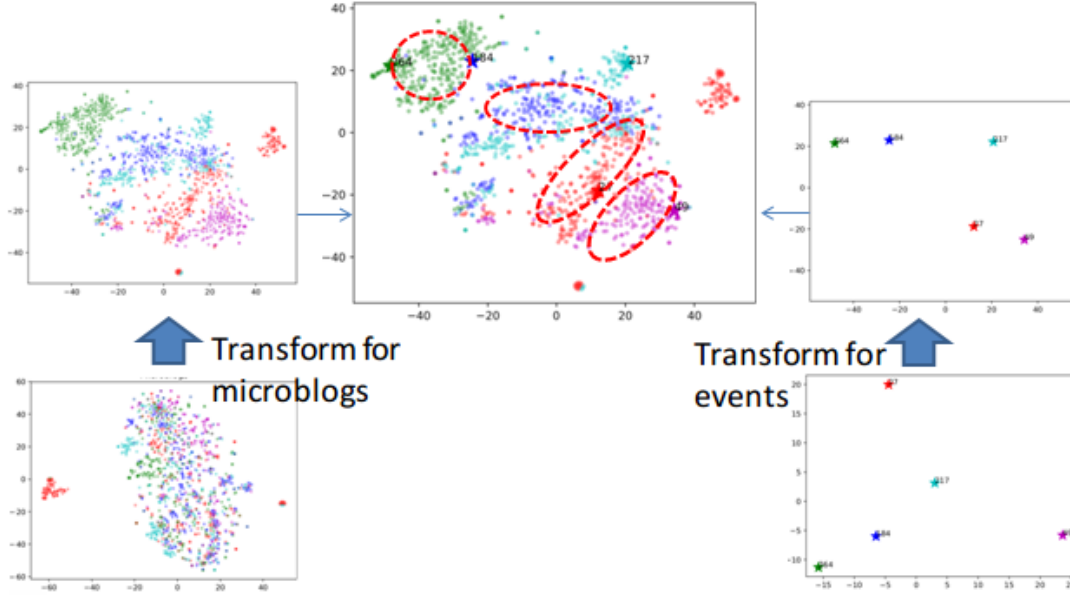


Figure 5.11: Visualization of how dual transformations worked.

5.5.4 Visualization of How Dual Transformations Worked

Figure 5.11 showed the visualization results of dual transformations. Before transformation, the microblogs referring to the same events were not close each other, which could be more distant for those microblogs of implicit references. After transformation, microblogs referring to same events could be close and thus this could be useful for identifying their referred events. DT furthermore transformed for events, such that the transformed events could be close to their relevant microblogs, in the space after transformation. Note that most of competitive models can be seen as only transforming from one-side (such as microblogs), for their matching tasks.

5.6 Discussion

We listed the findings from the experimental results as follows.

- **Deep transfer learning with dual transformations (DT) was effective to identify implicitly or explicitly referred events.** It was also applicable to unseen events that did not exist in training data.
- **DT outperformed baseline methods.** DT was better than single-direction transformations under multiple modeling. DT was better than the method of Chapter 4, which used similar links. Corpus-based was useful on explicit references. Word-to-word mapping (B2E) did not work.
- **Dense modeling or predicting reference type improved on implicit references.** Applying DT with dense modeling or reference-type-based DT could improve on implicit references. Applying with CNN-based models had limited improvement.
- **DT was robust when features were incomplete** and we should model DT with rich features. In the results, DT with full features was better than DT without the features of location and time, and followed by DT with TFIDF features.

We further discussed the limitation or future work as follows.

- **Transformations for more diverse microblogs and events.** In our modeling, we learned one instance of transformations model. However, one general instance of transformations might not meet diverse microblogs and events for the matching task. Especially, the contexts of microblogs and events in real world, such as term meaning and expressions, changes dynamically in different regions or in timeline. We should apply and learn suitable transformations accordingly.

CHAPTER 6

6. SEARCHING FOR MICROBLOGS REFERRING TO EVENTS BY DEEP DYNAMIC QUERY STRATEGIES

6.1 Introduction

Searching for microblogs referring to a particular event is useful when a user intends to survey people's opinions and feelings that are not reported in news articles. Most studies of event-focused microblog searching tend to heavily rely on the event contents, which are often keywords or named entities of events. However, this approach is unsuitable when the reference to the event is implicit and excludes event contents. We discuss the challenging of this microblog search problem as follows. (1) Searching for microblogs implicitly or explicitly referring to events. For example, a user may describe the result of a New York Yankees vs. Boston Red Sox game as "A really disappointing game." In this case, search keywords such as "yankees red sox" would not find such microblogs, while more general keywords such as "game" would produce too many search results. (2) The limited search results provided by the search engine for microblogs. A search engine usually has its own ranking criteria for search results or has the limited number of return. For example, even if a user makes effective queries for text contents but he/she cannot get the relevant microblogs posted in the past or specific regions because the search engine prefers the latest or global ones in the top of search results. Furthermore, as the

6. Searching for Microblogs Referring to Events by Deep Dynamic Query Strategies

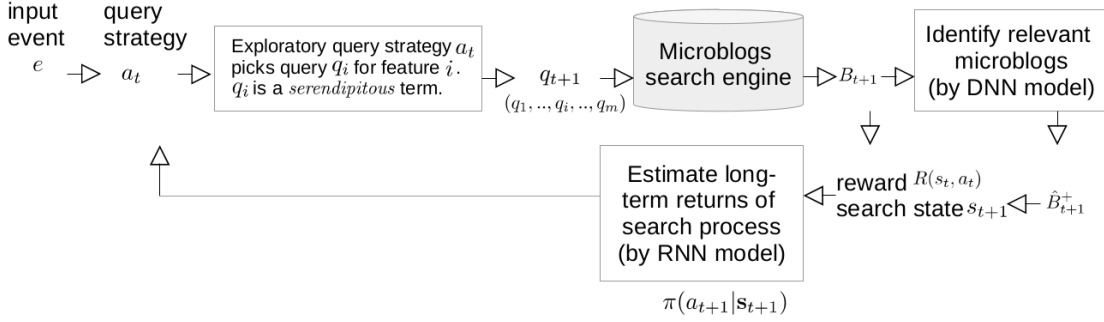


Figure 6.1: Our search process dynamically apply exploratory or exploitative query strategies to search for microblogs referring to event e , through search engine. We select query strategy a_t by policy $\pi(a_t|s_t)$ and generate query q_{t+1} , and retrieve microblogs B_{t+1} , identify the relevant microblogs $\hat{B}_{t+1}^+$, and transit to the next search state s_{t+1} .

fraction of microblogs referring to a particular event can be extraordinarily small in many cases, finding effective queries under the API call limit imposed by the microblog hosting company is a challenging task for both humans and algorithms. (3) Last but not least, we maximize search efficiency, which is measured by the recall of relevant microblogs for a fixed amount of issued queries.

To this end, we propose a dynamic search process that takes query strategies optimized for the current search state. This approach monitors the search states and takes the most effective query strategy for the next step. This approach consists of the following three components. (a) For this microblog search problem, we propose exploratory and exploitative query strategies to search for microblogs implicitly or explicitly referring to events. (b) To accurately monitor the search process, we propose a deep Q-learning approach with RNN model for predicting long-term returns of search process by using sequential search states and query strategies. (c) To increase the applicability of this research, i.e., the search process can be applied to not only events seen during training phase but also to unseen or new events, we optimize online query strategies by estimating the relevance of microblogs using the proposed DNN model in Chapter 5.

The motivation of applying reinforcement learning [72] is explained as follows. For the microblog search problem, the given search engine (environment) can be unknown to the proposed search process (agent). For example, the ranking criterion of search results from the engine is unknown to the search process. Meanwhile, as the search results are usually limited and the fraction of relevant microblogs can be extraordinarily small,

6. Searching for Microblogs Referring to Events by Deep Dynamic Query Strategies

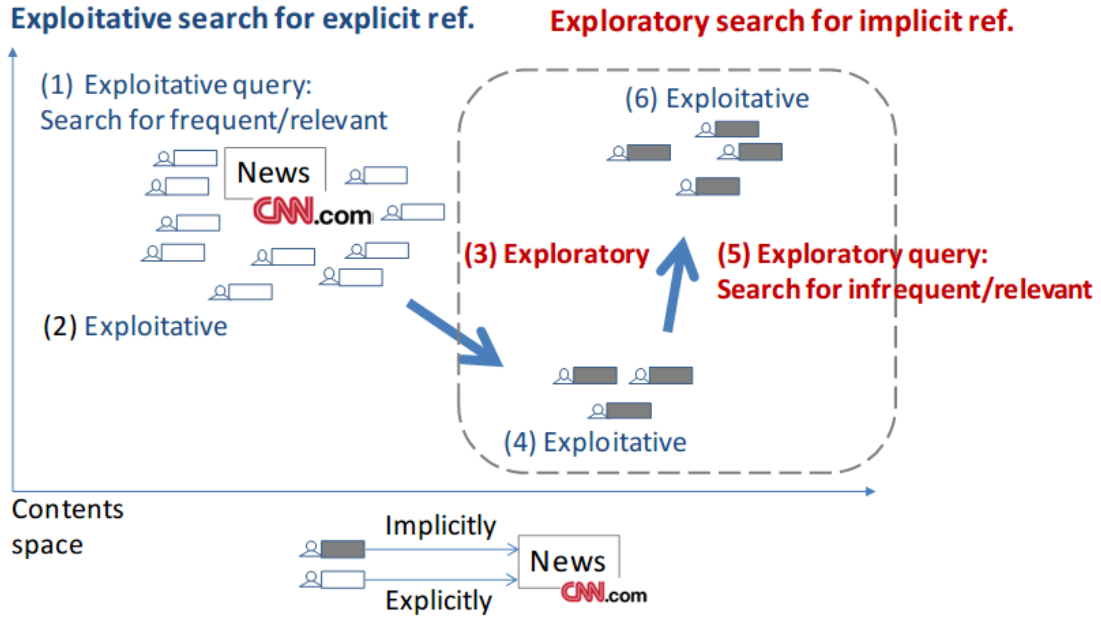


Figure 6.2: We propose exploratory and exploitative query strategies to search for microblogs implicitly or explicitly referring to events.

searching for relevant microblogs on the engine (exploring the environment) can be much difficult. Therefore, we apply reinforcement learning to assist the search process to take effective query strategies (actions) based on the observed search states (states) and the predicted long-term returns (returns).

We conducted simulation-based experiments using 31,174 tweets randomly sampled from Twitter. The main experimental results and the contributions of this work are summarized as follows:

- We propose a robust reinforcement learning framework with exploratory and exploitative query strategies that can search for microblogs implicitly or explicitly referring to events.
- We propose a deep Q-learning approach with RNN model for predicting long-term returns of search process by using sequential search states.
- We optimize online query strategies by estimating the relevance of microblogs using the estimator of Chapter 5.
- We demonstrated the effectiveness of our search process for both references and the target of unseen events.

6. Searching for Microblogs Referring to Events by Deep Dynamic Query Strategies

The rest of this research is organized as follows. We show the motivation and novelty compared with previous studies in Section 6.1. The microblog search problem is presented in Section 6.2. We propose a dynamic search process in Sections 6.3, 6.4, and 6.5. We evaluated the proposed methods using real-world microblog data and the target of unseen events, discussed the experimental results, and presented a case study in Sections 6.7 and 6.8.

Table 6.1: Microblog search problem and related works.

	Problem: search for microblogs implicitly or explicitly referring to events through search engine	Problem: search for microblogs (documents) explicitly referring to events (keyword-based)	Problem: search for documents by following links in documents
Ours (Ch 6)	o	o	
[64, 65, 67-70, 73, 74]		o	
[71, 117-119]			o

6.1.1 Motivation

Searching for microblogs implicitly or explicitly referring to events

We are few or novel ones to address the problem of searching for microblogs implicitly or explicitly referring to events, through search engine. Note that the microblogs of implicit references do not contain any literal terms of events, while previous works [64, 65, 67-70, 73, 74] proposed to search for microblogs (or documents) explicitly referring to events (or targets), where those microblogs contain literal terms of events. Another related works utilized the links in documents for search [71, 117-119]. With the limited scope of searching for relevant microblogs, either by explicit references or by static links, the previous works might not find more diverse microblogs referring to events.

Searching for diverse microblogs through search engine, without spatial or temporal limitation

Note that the above related works [64, 65, 67-71, 73, 74, 117-119] did not utilize typical search engines for search, and most of them directly searched in a given set of microblogs (or documents). Some works [7, 25, 69, 70] searched for local or real-time microblogs, where the search results can be limited by spatial or temporal domains. In our problem, we search for microblogs through search engine by queries that include textual contents, writer profiles, posted locations or time. We thus are able to retrieve more, diverse, and past or latest search results provided by search engine. Note that the set of searchable microblogs by search engine can be dynamic, compared with a given dataset.

A recall-oriented search task

The challenging of the microblog search problem is as follows. (1) Searching for microblogs implicitly or explicitly referring to events is a low-recall task and keyword-based approaches may not work. As the case introduced in Chapter 6.1, keywords cannot find those microblogs without containing event contents and general words (e.g., game) find too much redundant results. (2) The limited search results provided by search engine. In our problem, the ranking criterion of search engine is unknown. For example, even if we makes effective queries for textual contents but we cannot get relevant microblogs posted in the past time or specific regions because the search engine prefers the latest or global ones as search results. Moreover, the number of one-time search results and the number of issuable queries for a search engine can be limited, especially for commercial search engines. (3) Therefore, we propose to maximize the search efficiency, which is measured by recall for a fixed amount of issued queries.

6. Searching for Microblogs Referring to Events by Deep Dynamic Query Strategies

Table 6.2: Method comparison. We compare on (deep) reinforcement learning frameworks.

	Reinforcement learning framework									
	Optimize online query strategies	Exploratory actions (query strategies)	Use sequentially past experiences	RNN models	Reduce overesti- mation	Exploitative actions (including keyword-based, predefined-based)	DL models	Learning Q- learning	Policy Gradient	Other learning
Ours (Ch 6)	o	o	o	o	o	o	o	o		
[71, 74]						o		o		
[72]						o			o	
[73]						o				o
[116]			o	o		o	o	o		
[75, 76]					o	o	o	o		
[77]						o	o	o		

6.1.2 Novelty of Our Approach

A robust reinforcement learning framework with exploratory and exploitative query strategies

To search for microblogs implicitly or explicit referring to events, through search engine, we propose to utilize exploratory and exploitative query strategies in our reinforcement learning framework. In contrast, in common reinforcement learning frameworks of focus crawling [71-74] or playing Atari games [75-77, 116], their actions or states are limited or predefined and thus the ability of exploring in an unknown environment might be limited, which do not fit for our microblog search problem. For example, focus crawling [73] for documents utilized the exploitative types of query terms, which are effective for documents with keywords. However, for a low-recall search task, the already used exploitative types of terms cannot be useful since exploring more unseen documents or documents without keywords is important. Furthermore, compared with common frameworks, our reinforcement learning framework utilizes their advantages, including deep Q-learning with RNN model and the technique for reducing overestimation of Q-values.

We can optimize online query strategies for events that are unseen during training phase

One important issue is the search ability in online phase (testing phase), where the ground-truth data is not provided and thus the rewards of search results is unknown to the search process. In competitive works [71-77, 116], the reinforcement learning framework utilized the well-trained models for predicting reward values in online phase and deciding actions. However, their way is not suitable for our problem. For our low-recall search task, we encourage exploring more unseen microblogs or microblogs without keywords, which can be in different and diverse topics, and thus the model trained during training phase might not be useful in online search. Therefore, we propose to optimize online query strategies by predicting rewards using the estimated relevance of microblogs. Especially, we can identify relevant microblogs for those unseen or new events, which do not exist in training dataset, by using a DNN model (which is introduced in Chapter 5).

Deep Q-learning with RNN model, by using sequential search experiences

To accurately monitor the search process, we use deep Q-learning with RNN model [116] and use not only current search states but also the history of search states and query strategies. Note that for optimizing online query strategies, we update the loss function of this model to optimize Q-values when running in online search.

6.2 Problem Definition

We first introduce several terms that are important for defining the target problem.

Microblog: a microblog is posted by a user of a social media platform, such as a tweet posted by a Twitter user. A microblog often includes textual content, a post time, and a post location.

Event: a common definition of an event is "a thing that happens, for a specific time and place, and can be observed by human" [90, 91], such as news, holidays, disasters. In this research, we study on news, a common type of events that are referred to by microblogs.

Reference: a microblog b refers to an event e if the writer posted b to reply to e . Note that evidence of a reply can be a hyperlink or a reply-link from b to e . A reference can be either explicit or implicit. For example, a microblog "Congrats to Red Sox, the winner of the 2018 World Series Championship!" explicitly refers to the event "The Red Sox beat the Dodgers in the 2018 World Series", while another microblog "Many congrats to Bosox." implicitly refers to the same event.

Problem definition (Searching for microblogs referring to events): given an event e and a search engine for microblogs, the problem is to retrieve as many microblogs referring to e as possible by issuing queries to the search engine, where the number of search results for one-time query is k and k is extremely small compared to the number of all microblogs. The success of an algorithm in solving this problem is measured by the recall of relevant microblogs after issuing n queries.

6.3 Search by Exploratory and Exploitative Query Strategies

6.3.1 MDP-based Search Process

To efficiently search for microblogs referring to an given event, we propose a dynamic search process based on MDP [92]. An MDP is composed of states S , actions A , a transition function T , a reward function R , and a policy π . Let $s_t \in S$ be a state at time t . The process takes an action $a_t \in A$ from a probability distribution $\pi(a_t|s_t)$, and moves to the next state s_{t+1} as determined by the transition function $T(s_t, a_t)$, i.e., $s_{t+1} = T(s_t, a_t)$. The process receives a reward $r(s_t, a_t)$ by taking the action a_t at the state s_t . The goal of the MDP is to find the policy that maximizes the cumulative rewards.

The query strategies of a search process often include (but are not limited to) exploration, which devises queries for retrieving microblogs not retrieved in past queries, and exploitation, which continues to retrieve microblogs similar to those found by recent queries. The dynamics of the search process highly depend on both the given event and the current search state, such as the numbers of relevant and newly found microblogs in the latest search results. These characteristics of the search process can be well modeled by MDP. At each time step, the search process chooses a query strategy (or an action) by following the optimized policy. The chosen query strategy then generates a query to retrieve new search results. The retrieved results update the state and issue a reward based on the number of relevant results. Below, we explain how each component of an MDP can be instantiated in the dynamic search process.

States

The state $s_t = (\mathbf{d}_t, \mathbf{d}_t^+, \mathbf{d}_t^{rel}, \mathbf{d}_t^{new})$ of the search process at time t can be used to measure the degree of exploration and exploitation, i.e., whether the process is retrieving more unseen microblogs than familiar ones, or vice versa.

$\mathbf{d}_t$ represents the difference between the search results retrieved by a query at time t (denoted by $\mathbf{q}_t$), and the previous search results, i.e., those retrieved by $\mathbf{q}_{t-1}$. More precisely, let $\mathbf{b} = (b_1, b_2, \dots, b_m)$ be a vector representing m features of a microblog, where b_1 and b_2 are the textual content and post time of a microblog, respectively. The i -th value of $\mathbf{d}_t$ is defined as

6. Searching for Microblogs Referring to Events by Deep Dynamic Query Strategies

$$d_{t,i} = \|\bar{b}_{t,i} - \bar{b}_{t-1,i}\|, \quad (6.1)$$

where $\bar{b}_{t,i}$ is the mean of the microblogs B_t for the i -th feature: $\bar{b}_{t,i} = \frac{1}{|B_t|} \sum_{b \in B_t} b_i$, and B_t is a set of microblogs retrieved by the query at time t .

$\mathbf{d}_t^+$ represents the difference between the relevant search results at time t (denoted by q_t), and the previous search results, i.e., those retrieved by q_{t-1} . More precisely, let $b = (b_1, b_2, \dots, b_m)$ be a vector representing m features of a microblog, where b_1 and b_2 are the textual content and post time of a microblog, respectively. The i -th value of $\mathbf{d}_t^+$ is defined as:

$$d_{t,i}^+ = \|\bar{b}_{t,i}^+ - \bar{b}_{t-1,i}^+\|, \quad (6.2)$$

where $\bar{b}_{t,i}^+ = \frac{1}{|B_t^+|} \sum_{b \in B_t^+} b_i$, and B_t^+ is the set of relevant microblogs retrieved by q_t .

d_t^{rel} is the difference between the number of relevant search results at time t , and the number of relevant search results at time $t-1$:

$$d_t^{rel} = |B_t^+| - |B_{t-1}^+|. \quad (6.3)$$

Finally, d_t^{new} is the number of newly found search results n_t at time t :

$$d_t^{new} = |B_t - \cup_{t'=1}^{t-1} B_{t'}|, \quad (6.4)$$

Note that $\mathbf{d}_t^+$ and d_t^{rel} are estimated except in the training phase when the relevant microblogs are known. When the ground truth is unavailable, the relevant microblogs can be estimated by our proposed DNN-based model as described in Chapter 5.

Actions

Our dynamic search process performs two types of actions, namely, exploitative and exploratory query strategies.

1. **Exploitative query strategy:** This type of strategies uses a salient term or a central value in the latest search results as a query. For example, a query strategy uses the content term with the maximum term frequency-inverse document frequency (TF-

6. Searching for Microblogs Referring to Events by Deep Dynamic Query Strategies

IDF) score, or a post time close to the average post time in the latest search results as a query.

2. Exploratory query strategy: This type of strategies uses a serendipitous term or an outer value in the latest search results as a query. For example, a query strategy uses the content term with a low TF-IDF score, or a post time far from the average post time in the latest search results as a query.

We can define both types of query strategies for each feature of a microblog. Therefore, the number of actions A is $2m$, where m is the number of features.

Transition

In the dynamic search process, a transition refers to a change from search state s_t to search state s_{t+1} , and is determined by the search results B_t , B_t^+ , B_{t+1} , and B_{t+1}^+ .

Reward

A reward is an immediate return of executing an action. In our method, the reward is the number of relevant microblogs in the search results; that is, $R(s_t, a_t) = |B_{t+1}^+|$. As the relevant microblogs are not known *a priori*, the relevance of the retrieved microblogs is estimated by the DNN-based model proposed in Chapter 5.

Policy

Policy describes the behaviors of the dynamic search process. Given the current state s_t and event e , the policy $\pi(a_t|s_t)$ selects an action $a_t \in A$ based on the long-term return of the search process.

Our search process is demonstrated in Figure 6.1. Given an event e , we select a query strategy a_t based on the policy $\pi(a_t|s_t)$ that considers the long-term return of applying a_t according to search states s_t , and generate query q_{t+1} for the next time. We then retrieve microblogs B_{t+1} and estimate the reward $|\hat{B}_{t+1}^+|$, and transit to the next search state s_{t+1} .

6.3.2 Exploratory and Exploitative Query Strategies

As mentioned, there were two types of query strategies for each feature of a microblog.

Exploitative query strategies for a textual content and a writer's profile issue the term v that maximizes the following formula as a query at time $t + 1$:

$$TF-IDF(v, B_t)TF-IDF(v, B_t^+), \quad (6.5)$$

where the function $TF-IDF(v, B_t)$ indicates the weight of v in B_t by the TF-IDF weighting schema. Recall that B_t is a set of microblogs retrieved at time t , and B_t^+ is a set of relevant microblogs retrieved at time t . Exploitative query strategies for a post location and a post time issue the location or time of the microblog randomly sampled from the five microblogs closest to the mean post location or post time in search results B . By taking an example of the post time, the closeness of a microblog b_i is computed by:

$$|b_{i,z} - \bar{b}_z|, \quad (6.6)$$

Where $\bar{b}_z$ is the mean post time in search results B_t . The closeness in terms of the post location is defined in a similar way with the Euclidean distance.

Exploratory query strategies

In contrast to the exploitative query strategies that encourage retrieval of microblogs similar to the latest search results, exploratory query strategies seek microblogs that are dissimilar to the latest search results. Exploratory query strategies for a content and a writer's profile issues the term v that maximizes the following formula as a query at time $t + 1$:

$$TF-IDF(v, B_t)^{-1}TF-IDF(v, B_t^+). \quad (6.7)$$

Only the difference from the exploitative query strategies is the inverse weight by $TF-IDF(v, B_t)$. This is expected to encourage less frequent but salient terms to be issued as a query. Exploratory query strategies for a post location and a post time issue the location or time of a microblog that is randomly sampled from the five microblogs farthest from the mean post location or post time in the search results B .

6. Searching for Microblogs Referring to Events by Deep Dynamic Query Strategies

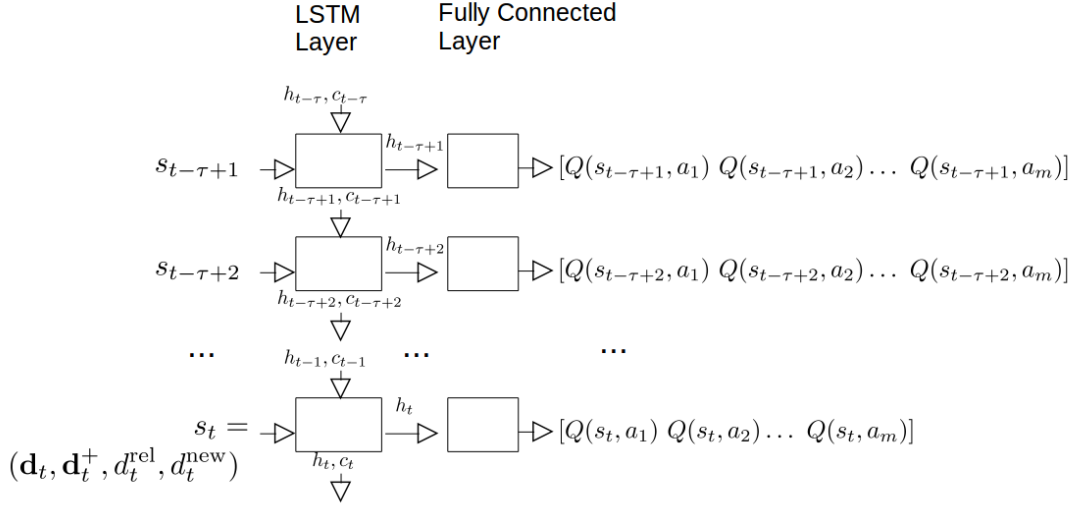


Figure 6.3: Our RNN-based model predicts the long-term returns $Q(s_t, a)$ of each query strategy $a \in A$ by using a sequence of search states $s_t = (s_{t-\tau+1}, s_{t-\tau+2}, \dots, s_t)$.

6.4 Deep Q-learning with RNN Model

As explained above, the goal of the MDP is to learn the policy that maximizes the cumulative rewards of a search process. By the Q-learning algorithm [75-77], the long-term return of a search process that issues query strategy a at time t , under policy π , is defined as the sum of discounted cumulative rewards,

$$Q(s_t, a_t) = \mathbb{E}[R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \dots | s_t, a_t, \pi]. \quad (6.8)$$

Meanwhile, the Q function can determine the policy π :

$$\pi(a_t | s_t) = \frac{\exp(Q(s_t, a_t))}{\sum_{a \in A} \exp(Q(s_t, a))}. \quad (6.9)$$

Intuitively, the policy prefers actions with high long-term returns and we find the optimal policy by learning the Q function. By the approach of temporal difference learning in Q-learning, the Q function is learned so that the following loss function is minimized:

$$\mathbb{E}[(R(s_t, a_t) + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t))^2], \quad (6.10)$$

where $\gamma \in [0, 1]$ controls the importance between the immediate and future returns. Under this loss function, the Q function should approximate the long-term return

6. Searching for Microblogs Referring to Events by Deep Dynamic Query Strategies

$Q(s_t, a_t)$ of taking action a_t at time t by using the target, $R(s_t, a_t) + \gamma \max_a Q(s_{t+1}, a)$, which is the sum of the immediate reward $R(s_t, a_t)$, and the future return of the maximum value of the Q function at the next state s_{t+1} .

The RNN model

In standard Q-learning, the long-term reward is estimated only from the current state and action, not by the past states and actions. To overcome this drawback, we propose an RNN-based model in which the Q function estimates the long-term return from the history of states and actions. The Q function with this model determines the policy $\pi(a_t|s_t)$ by considering the past states $s_t = (\dots, s_{t-1}, s_t)$, and is denoted by $Q(s_t, a)$. The RNN model comprises a long short-term memory (LSTM) [93] layer and a fully connected (FC) layer, which is detailed in the next section.

The RNN model is illustrated in Figure 6.3. A sequence of states $s_t = (s_{t-\tau+1}, s_{t-\tau+2}, \dots, s_t)$ of length τ is input to the LSTM layer of h -dimensional hidden states. Inputting the hidden state at time t to a fully connected layer with a $h \times |A|$ matrix, we obtain the long-term reward $Q(s_t, a)$ at time t for all $a \in A$.

Double Q-learning and experience replay

The Q function is trained by a deep Q-learning algorithm. For clarity, we denote by $Q(s, a; \mathbf{W})$ the Q function parameterized by $\mathbf{W}$. We then update Q by the following iterative process:

1. In each iteration, choose a t at each time t by an ϵ -greedy selection method. This method either randomly selects a t with probability ϵ or $a_t = \underset{a}{\operatorname{argmax}} Q(s_t, a; \mathbf{W})$ with probability $1 - \epsilon$.
2. Compute the corresponding reward $R(s_t, a_t)$ of executing action a_t , and move to the next state s_{t+1} .

A large ϵ encourages exploration of various actions, whereas a small value encourages the selection of the best action for the current Q function.

Here we adopt experience replay [75], a common technique in deep Q-learning that randomly chooses from the previously observed state-action sequences and updates the Q function by the sampled sequences. We additionally employ double Q-learning [76] for stable learning. The current model selects the actions and another model, which is the

6. Searching for Microblogs Referring to Events by Deep Dynamic Query Strategies

previously trained one, determines the value of the Q function under the selected actions. Double Q-learning is used to reduce the overestimations of the estimated Q-values. More formally, the loss function is now defined as:

$$L_q(\mathbf{W}) = \mathbb{E}[(R(s_t, a_t) + \gamma Q(s_{t+1}, a_{t+1}^*; \mathbf{W}') - Q(s_t, a_t; \mathbf{W}))^2], \quad (6.11)$$

where $a_{t+1}^* = \arg\max_a Q(s_{t+1}, a; \mathbf{W})$. Note that this equation involves two different models. The model with $\mathbf{W}$ selects the action a_{t+1}^* , whereas the model with $\mathbf{W}'$ computes the value of the Q function for a_{t+1}^* . The parameter set $\mathbf{W}'$ is updated to $\mathbf{W}$ after every constant number of iterations. Therefore, we can learn the optimal Q function with the gradient

$$\nabla_{\mathbf{W}} L_q(\mathbf{W}) = \mathbb{E}[(R(s_t, a_t) + \gamma Q(s_{t+1}, a_{t+1}^*; \mathbf{W}') - Q(s_t, a_t; \mathbf{W})) \nabla_{\mathbf{W}} Q(s_t, a_t; \mathbf{W})], \quad (6.12)$$

and we optimize by stochastic gradient descent.

6.4.1 Predicting Long-Term Returns by RNN Model

We propose an RNN-based model that is used to implement the Q function $Q(s, a; \mathbf{W})$ that computes the long-term returns for each query strategy $a \in A$.

LSTM layer

To this end, the RNN model comprises an LSTM layer and a fully connected (FC) layer. The LSTM layer [94, 95] can remember previous search states and keeps track of the dependencies among the search states, for arbitrary and long time-intervals. The FC layer then transforms the output of the LSTM, which is the information of the required current and past search states, into the long-term returns of each query strategy a .

6. Searching for Microblogs Referring to Events by Deep Dynamic Query Strategies

In Figure 6.3, given our search state s_t at time t , s_t is first input to the LSTM layer with the hidden state h_{t-1} and memory cell state c_{t-1} [94]:

$$\begin{aligned} f_t &= \sigma(\mathbf{W}_f s_t + \mathbf{U}_f h_{t-1} + \boldsymbol{\delta}_f), \\ i_t &= \sigma(\mathbf{W}_i s_t + \mathbf{U}_i h_{t-1} + \boldsymbol{\delta}_i), \\ o_t &= \sigma(\mathbf{W}_o s_t + \mathbf{U}_o h_{t-1} + \boldsymbol{\delta}_o), \\ c_t &= f_t \otimes c_{t-1} + i_t \otimes \tanh(\mathbf{W}_c s_t + \mathbf{U}_c h_{t-1} + \boldsymbol{\delta}_c), \\ h_t &= o_t \otimes \tanh(c_t). \end{aligned} \quad (6.13)$$

Activation functions are sigmoid function (σ) and hyperbolic tangent function ($\tanh$). $\otimes$ denotes the operation of element-wise product. LSTM updates the state c_t of the memory cell, which keeps tracking the dependencies among previous search states, by controlling the extent i_t to which the new value (from search state s_t) should be used and the extent f_t to which the existing cell value c_{t-1} should be remained, in the memory cell. The output of new hidden state h_t is computed based on the extent o_t to which the state c_t of the memory cell should be used.

FC layer

The computed h_t is then input to the FC layer to compute the long-term returns for each query strategy $a \in A$:

$$[Q(s_t, a_1) \dots Q(s_t, a_{|A|})] = \text{ReLU}(\mathbf{W}_a h_t + \boldsymbol{\delta}_a)^T, \quad (6.14)$$

where $\mathbf{W}_a$ is a matrix of dimension $|A| \times h$, and $\boldsymbol{\delta}_a$ is biases of dimension $|A| \times 1$. The FC layer uses the activation function of rectified linear unit (ReLU) [96]:

$$\text{ReLU}([x_1 \dots x_{|A|}]^T) = [\max(0, x_1) \dots \max(0, x_{|A|})]^T, \quad (6.15)$$

where $x_i \in \mathbb{R}$. Therefore, for the Q function $Q(\mathbf{s}, \mathbf{a}; \mathbf{W})$, $\mathbf{W}$ comprises the trainable parameters below. In LSTM layer, the parameters are the matrices $\mathbf{W}_f, \mathbf{W}_i, \mathbf{W}_o, \mathbf{W}_c \in \mathbb{R}^{h \times s}$, the matrices $\mathbf{U}_f, \mathbf{U}_i, \mathbf{U}_o, \mathbf{U}_c \in \mathbb{R}^{h \times h}$, and the biases vectors $\mathbf{U}_f, \mathbf{U}_i, \mathbf{U}_o, \mathbf{U}_c \in \mathbb{R}^{h \times 1}$, where s is the dimension of search state s_t and h is LSTM dimension. In FC layer, the parameters are $\mathbf{W}_a$ and $\boldsymbol{\delta}_a$.

6.5 Optimizing Online Query Strategies

In our dynamic search process, the reward is defined as the number of relevant search results, and the relevance is assumed to be known in the previous subsection. When the dynamic search process is used in testing phase for unseen events, i.e., no ground-truth relevance is known, the immediate reward of the query strategies is estimated using the relevant microblogs identified by the DNN model. We regard a microblog b as relevant to a given event e if the score $F(e, b)$ is sufficiently large. An estimated set of relevant microblogs is formally described as

$$\hat{B}_t^+ = \{b | b \in B_t \wedge F(b, e; \mathbf{W}) \geq \rho\}, \quad (6.16)$$

where ρ is the score threshold. The immediate reward then can be computed by the same procedure as when the relevant microblogs are known *a priori*. Therefore, the Q function that computes the long-term returns can be optimized when online searching by the updated loss function:

$$L_q(\mathbf{W}) = \mathbb{E}[(\hat{R}(s_t, a_t) + \gamma Q(s_{t+1}, a_{t+1}^*; \mathbf{W}') - Q(s_t, a_t; \mathbf{W}))^2], \quad (6.17)$$

where $\hat{R}(s_t, a_t) = |\hat{B}_{t+1}^+|$.

6.6 Experimental Setting

We explained the experimental setting including the dataset collection and annotation, simulated search engine and queries, comparative methods, the dataset setting for evaluation using unseen events, and the tuning of our dynamic search process.

6.6.1 Dataset Collection

For the microblog search experiments, we collected 31,174 tweets (microblogs) from Twitter, among which 16,535 tweets referred to a specific news (event) and the others were randomly selected. Please refer to Section 5.4.1 for the details of the dataset collection, the check of identity of events, and preprocess for microblogs' locations.

6.6.2 Search Engine and Query

To simulate a search process, we built a simple search engine that returns the top-k microblogs according to the following matching score:

$$\sum_{i \in \{c, u, l, z\}} \alpha_i \text{sim}_i(\mathbf{q}_i, \mathbf{b}_i). \quad (6.18)$$

Note that k was set to 90 in the experiments. The query vector $\mathbf{q} = (\mathbf{q}_c, \mathbf{q}_u, \mathbf{q}_l, \mathbf{q}_z)$ was constructed in a similar way to the microblog vector:

$$\begin{aligned} \mathbf{q}_c &= \frac{1}{|V_c|} \sum_{v \in V_c} WE_c(v), \\ \mathbf{q}_u &= \frac{1}{|V_u|} \sum_{v \in V_u} WE_u(v), \end{aligned} \quad (6.19)$$

where v is either a query term for the contents or a query term for writer's profiles. The similarity function sim_i can be different. When b_i is either a content vector or a writer-profile vector, sim_i is the cosine of the two vectors. The similarity function computes the inverse of the Euclidean distance for location and time. α_i was set based on the simulation conditions and was set as equal weight by default.

6. Searching for Microblogs Referring to Events by Deep Dynamic Query Strategies

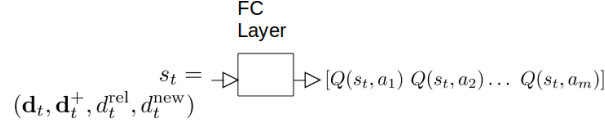


Figure 6.4: A model predicts long-term returns of query strategies by only the current search state.

6.6.3 Comparative Methods

We compared our method with the following methods from existing studies.

(1) Reinforcement learning with current search states (No RNN) [71-77]:

We evaluate the effect of our proposed RNN model on the search task. That is, the effect of RNN model for the deep Q-learning algorithm. We therefore compared with another DNN-based model (No RNN) that predicts the long-term returns of query strategies by using only the current search state as shown in Figure 6.4.

(2) Effect of relevance estimation on search ($TRANS_b$, $TRANS_e$, No WE) [87, 89, 111-115]:

We evaluate the effect of identifying relevant microblogs on the search task. Note that the below works (3) and (4) are also relevance estimation methods. We therefore compare the following three models ($TRANS_b$, $TRANS_e$, and No WE) that have different ability of relevance estimation, on the search task. Note that ours use DT for relevance estimation and the details of models are described in Chapter 5.

6. Searching for Microblogs Referring to Events by Deep Dynamic Query Strategies

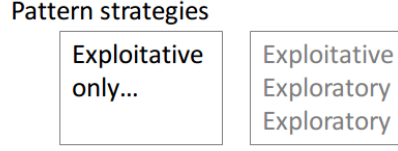


Figure 6.5: Pattern query strategies.

(3) **Blog-to-event-based search (B2E)** [102-104]:

This approach finds the clues for relevant microblogs by utilizing the mapping possibility from terms of microblogs into terms of events in sequence. Since an event and its associated microblogs have different vocabularies from different domains, this B2E approach utilizes a sequence-to-sequence encoder-decoder model as shown in Figure 5.3. With having the mapped terms in event's domain from microblogs, the relevant microblogs can be re-ranked as follows:

$$B'_{t-1} = \{b | b \in B_{t-1} \wedge \text{sim}(\mathbf{e}, \psi(b_c)) \geq \theta_{B2E}\}, \quad (6.20)$$

where the model ψ is trained by a corpus D including the word embeddings.

(4) **Corpus-based search (CW, CS)** [68, 73, 97-101]:

This approach is an intuitive way to find relevant documents based on the correlation of contexts of terms and their occurrence frequencies. Also, the word embeddings models are applied here. A simple search problem might be tackled with this approach, especially with enough issuing queries. In this category of methods, the search queries are generated using the similarities and weights of the terms in event $\mathbf{e}$, the search results B , and a corpus D . The first method (denoted by CW) uses the terms with the high frequency in the corpus and the latest search results to generate the next query $\mathbf{q} = (\mathbf{q}_c, \mathbf{q}_u, \mathbf{q}_l, \mathbf{q}_z)$:

$$\begin{aligned} \mathbf{q}_c &= WE_c(\text{argmax}_v(\lambda_B f_{B_{t-1}}(v) + \lambda_D f_D(v) + \lambda_n f_n(v))), \\ \mathbf{q}_u &= \frac{1}{|B_{t-1}|} \sum_{b \in B_{t-1}} \mathbf{b}_u, \\ \mathbf{q}_l &= \frac{1}{|B_{t-1}|} \sum_{b \in B_{t-1}} \mathbf{b}_l, \\ \mathbf{q}_z &= \frac{1}{|B_{t-1}|} \sum_{b \in B_{t-1}} \mathbf{b}_z, \end{aligned} \quad (6.21)$$

6. Searching for Microblogs Referring to Events by Deep Dynamic Query Strategies

where $f_{B_{t-1}}(v)$, $f_D(v)$, and $f_n(v)$ are the frequency of word v in B_{t-1} , the frequency of v in D , and the inverse count of the queries that yielded search results including v , respectively. The function $WE_c(v)$ indicates the word embedding of v , and $\mathbf{q}_u$, $\mathbf{q}_l$, and $\mathbf{q}_z$ are the means of the respective vectors in B_{t-1} .

The second method CS uses the same query vector as the CW, but a different $\mathbf{q}_c$. Intuitively, $\mathbf{q}_c$ is the vector representation of the most frequent term in the retrieved microblogs semantically similar to a given event e . In the CS, it is defined as

$$\begin{aligned}\mathbf{q}_c &= WE_c(\operatorname{argmax}_v f_{B'_{t-1}}(v)), \\ B'_{t-1} &= \{b | b \in B_{t-1} \wedge \operatorname{sim}(\mathbf{e}, \mathbf{b}_c) \geq \theta_{CS}\},\end{aligned}\tag{6.22}$$

where $\operatorname{sim}(\mathbf{e}, \mathbf{b}_c)$, which should approximate the semantic similarity of e and b , is computed as the cosine of their vector representations trained by a corpus D . The hyperparameter θ_{CS} is the threshold that determines whether a microblog is semantically similar to an event.

(5) Pattern strategies for search (PS, SS, Random):

For a search problem based on strategies for search (note that this microblog search problem is an example), we check whether the optimal or dynamic strategies are necessary for the problem. Therefore, we compare with the simple query strategies of repetitively issuing good patterns. We use one repetitive pattern (PS), which iteratively issues the exploratory and then the exploitative query strategies for contents, and another pattern (SS), which contains only the exploratory query strategy for contents. Note that we also check random strategies as a baseline.

6. Searching for Microblogs Referring to Events by Deep Dynamic Query Strategies

Table 6.3: The dataset setting for training and testing. Note that the events used in training and testing were disjoint.

Training phase	Testing phase (Online Searching)
# of seen events $ E_{tr} $: 85	# of unseen events $ E_{test} $: 23 (Note $E_{tr} \cap E_{test} = \emptyset$)
# of microblogs referring to events E_{tr} : 12,875	# of microblogs referring to events E_{test} : 3,660
	# of microblogs without reference: 14,639

6.6.4 Evaluation by Unseen Events

Using the dataset as shown in Table 6.3, in training phase, our and comparative models were trained by 80% of the training data and tuned the hyper parameters by the remaining 20% of the training data. In testing phase, we used the events E_{test} that were not included in the training data, $E_{test} \cap E_{tr} = \emptyset$, which was also called unseen events. Moreover, we included the random microblogs without references for simulating the microblog search problem by a search engine. In testing phase, there was about 1% of all microblogs that referred to a specific event.

6. Searching for Microblogs Referring to Events by Deep Dynamic Query Strategies

Table 6.4: Hyper-parameter setting of our models.

		value
Q-learning	ϵ	0.15
	γ	0.80
	Experience replay size	100
RNN	Sequence length (τ)	5
	LSTM dimension (h)	60
	# of trainable parameters of $\mathbf{W}$	17,528
DNN	event dimension (event’s content)	204
	microblog’s content dimension	274
	writer’s profile dimension	321
	microblog dimension	598
	# of trainable parameters of $\mathbf{W}_e$ and δ_e	$204 \cdot 204 + 204$
	# of trainable parameters of $\mathbf{W}_b$ and δ_b	$204 \cdot 598 + 204$

Table 6.5: Tuning of the RNN model, showing the recall scores for different sequence lengths and LSTM dimensions.

RNN sequence length (τ)	LSTM’s h			
	h=60	h=40	h=50	h=100
1	0.59	0.561	0.57	0.532
2	0.534	0.566	0.558	0.527
3	0.601	0.52	0.576	0.558
4	0.544	0.56	0.603	0.587
5	0.623	0.598	0.589	0.544
6	0.579	0.534	0.571	0.589
7	0.53	0.564	0.556	0.563
8	0.585	0.585	0.588	0.527
9	0.587	0.579	0.54	0.55
10	0.575	0.591	0.551	0.579

6.6.5 Tuning of Deep Dynamic Search Process

Table 6.4 summarizes our hyper-parameter settings for our dynamic search process. Below, we explain how we decided these parameters. In Q-learning framework, the search process was encouraged to explore various query strategies (with not very small ϵ) and cared about the immediate rewards (with not very large γ). We also optimized the query strategies by Q-learning after acquiring some search experience (the size of experience replay was 1/5 of the number of total queries for a search task). We used the setting by pre-experiments and this setting might meet some properties of a search task. For the size of the RNN model, we tested on the varying lengths τ of search states and the

6. Searching for Microblogs Referring to Events by Deep Dynamic Query Strategies

corresponding LSTM dimension h that could maximize the recall scores. Note that the training of the RNN model was carried out with stochastic gradient descent with a commonly used learning rate 0.01.

Evaluation metrics

We used Recall to measure the effectiveness of a search process:

$$recall = \frac{1}{|E_{test}|} \sum_{e \in E_{test}} recall_e, \quad (6.23)$$

where $recall_e$ is the recall of the search process for event e .

6. Searching for Microblogs Referring to Events by Deep Dynamic Query Strategies

Table 6.6: Microblog search performance on each method.

Method	Recall	Implicit	Explicit
CW	0.259	0.124	0.135
CS	0.225	0.100	0.125
B2E	0.228	0.096	0.132
Ours	0.561	0.331	0.229
No RNN	0.374	0.190	0.184
TRANS _b	0.517	0.292	0.225
TRANS _e	0.489	0.267	0.222
No WE	0.437	0.201	0.236
PS	0.379	0.187	0.192
SS	0.318	0.180	0.137
Random	0.190	0.084	0.106

Table 6.7: Microblog search performance on each method (with varying fractions of ground-truth relevant microblogs).

Method	80% of rel. microblogs			60% of rel. microblogs			40% of rel. microblogs			20% of rel. microblogs		
	Recall	Imp.	Exp.	Recall	Imp.	Exp.	Recall	Imp.	Exp.	Recall	Imp.	Exp.
CW	0.242	0.118	0.125	0.254	0.118	0.136	0.266	0.120	0.147	0.242	0.111	0.129
CS	0.213	0.092	0.121	0.215	0.090	0.125	0.205	0.077	0.126	0.198	0.063	0.131
B2E	0.179	0.069	0.109	0.182	0.065	0.115	0.188	0.078	0.107	0.185	0.055	0.123
Ours	0.548	0.328	0.220	0.511	0.306	0.205	0.558	0.324	0.234	0.542	0.296	0.246
No RNN	0.369	0.177	0.192	0.361	0.175	0.185	0.371	0.172	0.199	0.332	0.145	0.188
TRANS _b	0.452	0.242	0.210	0.467	0.235	0.232	0.480	0.255	0.225	0.477	0.248	0.229
TRANS _e	0.437	0.236	0.201	0.434	0.230	0.204	0.451	0.240	0.210	0.471	0.238	0.233
No WE	0.409	0.203	0.207	0.403	0.196	0.207	0.413	0.207	0.206	0.388	0.186	0.202
PS	0.337	0.170	0.167	0.332	0.160	0.172	0.346	0.169	0.177	0.364	0.174	0.190
SS	0.280	0.172	0.108	0.275	0.166	0.109	0.289	0.163	0.126	0.266	0.166	0.099
Random	0.154	0.072	0.082	0.137	0.077	0.061	0.170	0.068	0.102	0.146	0.068	0.072

6. Searching for Microblogs Referring to Events by Deep Dynamic Query Strategies

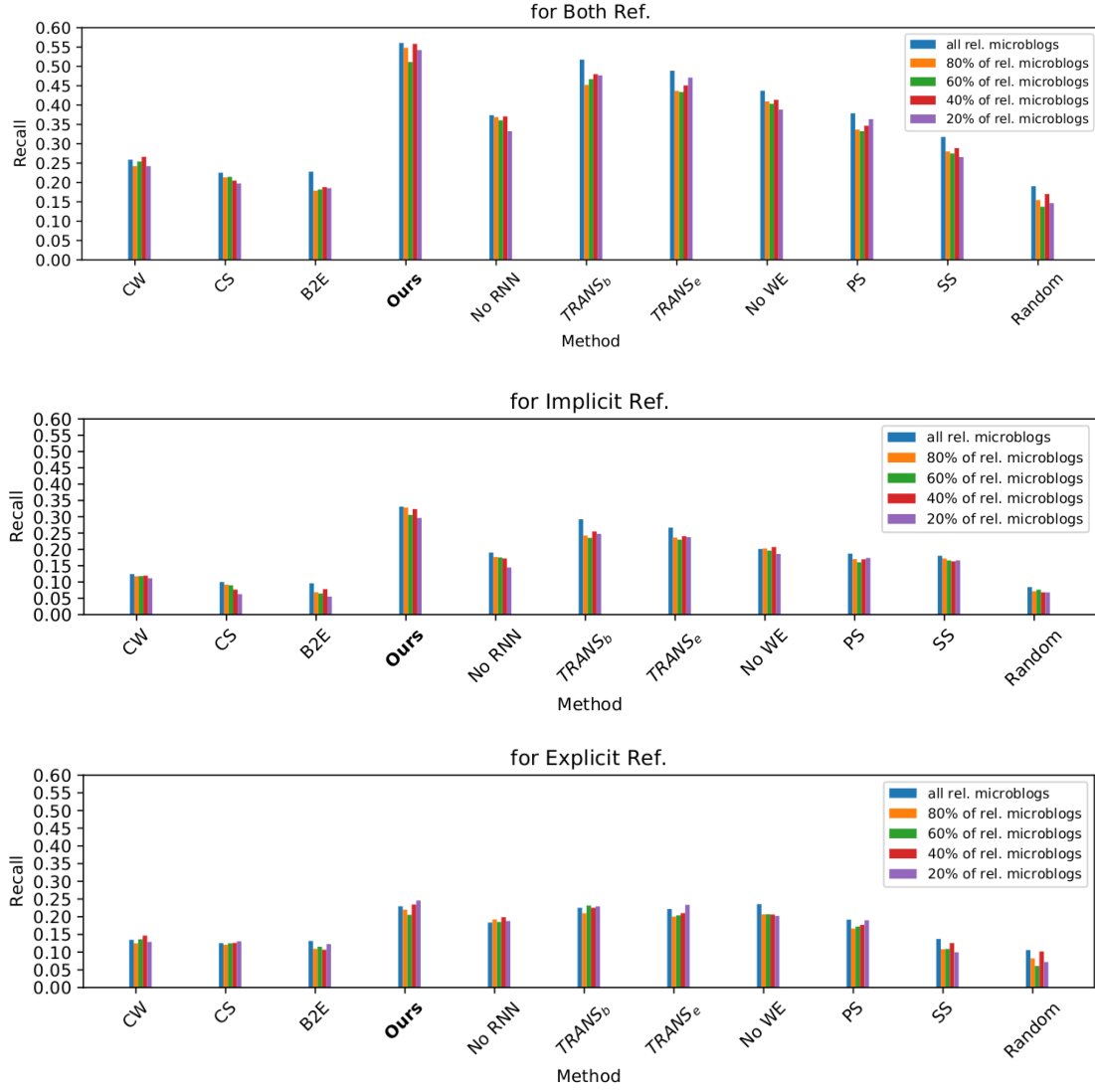


Figure 6.6: Microblog search performance on each method (with varying fractions of ground-truth relevant microblogs), for both references, implicit reference, and explicit references, respectively.

6.7 Experimental Evaluation

We answered these research questions through the extensive experiments.

- **RQ1.** Can our method effectively search for microblogs implicitly or explicitly referring to events? Can our method outperform competitive methods, such as search based on “reinforcement learning with current search states” and blog-to-event-based search? Is it also applicable to unseen events that did not exist in training data?
- **RQ2.** Can our method optimize online query strategies for a given unseen event?
- **RQ3.** Can the proposed RNN model improve the deep Q-learning?
- **RQ4.** Are our models robust? Do our models still work when the ranking criteria of search engine change? Do our models work when the query strategies are limited?

6.7.1 Search Performance with Varying Relevant Microblogs

Overall performance

Table 6.6 showed the overall performances of our proposed method and comparative methods. Our method (Ours) outperformed the baseline methods including the corpus-based search methods (CS and CW) and the blog-to-event-based search method (B2E), with at least 2.17 times higher recall. Our method achieved the best recall for implicit references and quality performance for explicit references, and ours was much effective for implicit references than explicit references. Applying reinforcement learning with RNN model was effective compared with the one without applying RNN model (No RNN). Applying word embeddings to represent microblogs or events was effective compared with the one without applying it (No WE). Applying dual transformations was effective than the one applying the single transformation either from microblogs ($TRANS_b$) or from events ($TRANS_e$). Furthermore, applying dynamic query strategies was effective than applying patterns (PS), (SS) of query strategies or random ones (Random). The corpus-based similarities by considering term frequencies (CW) was better than the blog-to-event-based (B2E) in implicit references.

6. Searching for Microblogs Referring to Events by Deep Dynamic Query Strategies

The main findings were listed as follows. (1) Our method was effective for implicit or explicit references in the search problem. (2) Applying dynamic query strategies was useful than patterns of strategies. (3) With dynamic query strategies, applying reinforcement learning and identifying relevant microblogs during a search process were both effective. (4) Measuring the semantic similarity by word embeddings and utilizing the mapping of words to words in sequence might not work for the search problem. We might recall the difficulty that relevant microblogs were extraordinarily small and the one-time search results were limited. (5) We might infer that an advanced way for identify relevant microblogs was required, especially for implicit references.

Search Performance with Less Relevant Microblogs

As our dataset contained a fair amount of relevant microblogs, we also evaluated the performances with the dataset including fewer relevant microblogs, which might better represent the distribution of real-world microblog data. Table 6.7 showed the recall of each method when a certain fraction of relevant microblogs was removed. Our method maintained its higher performance than the other methods, even when the relevant microblog ratio was reduced to 20%. However, the performance of implicit references decreased significantly, compared with that the performance of explicit references was not affected. The reason might be that finding implicit references required more relevant microblogs for guiding the search process.

We answered **RQ1** that our method effectively searched for relevant microblogs in implicit or explicit references, for unseen events, and ours outperformed each baseline method. Note that the events and their relevant microblogs used for evaluation were not included in the training dataset.

6. Searching for Microblogs Referring to Events by Deep Dynamic Query Strategies

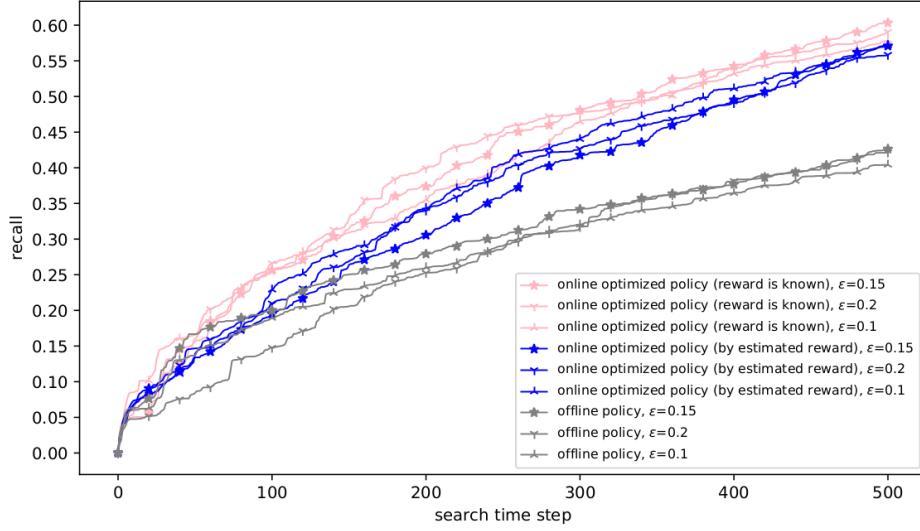


Figure 6.7: Effect of optimizing online query strategies. Blue: online optimized policy by estimated immediate rewards. Gray: offline policy without online optimization. Pink: online optimized policy by ground-truth rewards.

6.7.2 Effect of Optimizing Online Query Strategies

To check the effect of optimizing online query strategies, we compared with offline policy without online optimization, online optimization by estimated rewards (Ours), and online optimization by ground-truth rewards. Figure 6.7 showed that with around enough search experiences (around 150 search time steps), the optimized policies could guide the search well. Furthermore, optimizing by estimated immediate rewards could be close to the trend of the one optimized by ground-truth rewards. The findings were as follows. (1) Optimizing by estimated rewards worked. It could even achieve a quality performance without using a perfect relevance estimator, compared with the one using the ground-truth. (2) Even if immediate rewards were known, the search performance could not be perfect. This implied that there were other factors to tackle with the search problem. We answered **RQ2** that ours could optimize online query strategies for unseen events.

6.7.3 Effect of Deep Q-learning with RNN Model

To investigate the effect of applying the RNN model for deep Q-learning, we compared with the one using RNN model (Ours) and the one without RNN that predicted Q function by only current search states (No RNN). We visualized the comparison by projecting their found results into two dimensional spaces of textual contents, writer's profiles, locations, and times, respectively, by principal component analysis (PCA).

In Figure 6.8 (d), for the content space of event 209, most microblogs found by RNN or No RNN were overlapped as shown by the gray points. Meanwhile, these microblogs were in centers of microblogs and we could also find similar results on other events. This implied that either RNN or No RNN could find *representative* microblogs for events.

Furthermore, (1) RNN might find more *diverse* results, such as those of blue-cross points. In Figure 6.8 (a), for event 68: "President Trump put the onus of an 18-day partial government shutdown on Democrats ...", RNN found a microblog that replied implicitly with negative emotion: "i can see the *pain* ...", compared to a commonly found microblog: "But he was proud to take the mantle of a shutdown. Said he wouldnt blame the Democrats", which mentioned the keywords "shutdown" and "Democrats". RNN also found more results of diverse writer's profiles and locations, such as those in events 68 and 100. (2) Meanwhile, RNN exploited a bit more results that were around the center ones, such as in the content space of event 68.

(3) The above results might demonstrate the importance of *timely* exploratory or exploitative search with effective query features. Meanwhile, our RNN was effective to conduct *timely* exploratory and exploitative search by using the required historical search experiences, which might be the reason of the effectiveness of the RNN model. (4) Lastly, not only current search states but the historical search experiences could improve the search, which might be nature and was experimentally proved in this search problem. With the performance of RNN model (Ours) outperformed the one without RNN model (No RNN) and this case study analysis, we answered **RQ3** that our RNN model improved the deep Q-learning with using search experiences.

6. Searching for Microblogs Referring to Events by Deep Dynamic Query Strategies

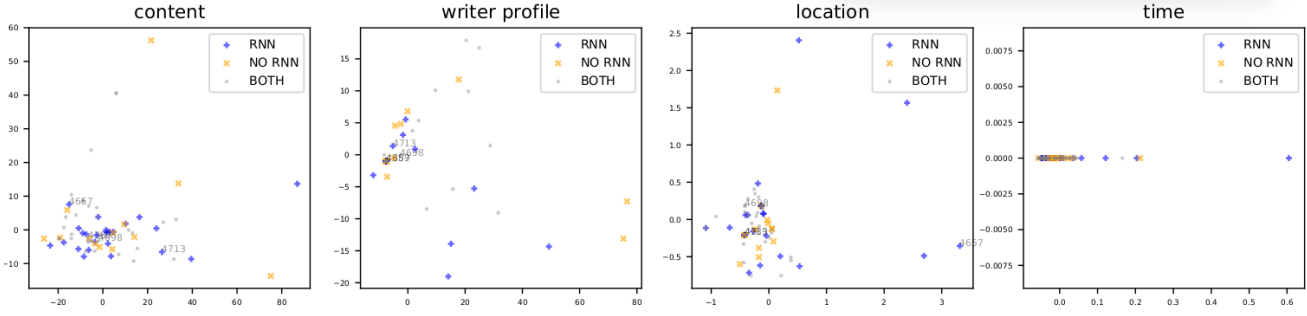


Figure (a): Relevant tweets (microblogs) found for event 68: "President Trump put the onus of an 18-day partial government shutdown on Democrats, saying the matter ...". A center tweet 4698 (which was from the center of tweets): "But he was proud to take the mantle of a shutdown. Said he wouldn't blame the Democrats. Lie?". A non-center tweet 4713: "i can see the pain. Use lube next time."

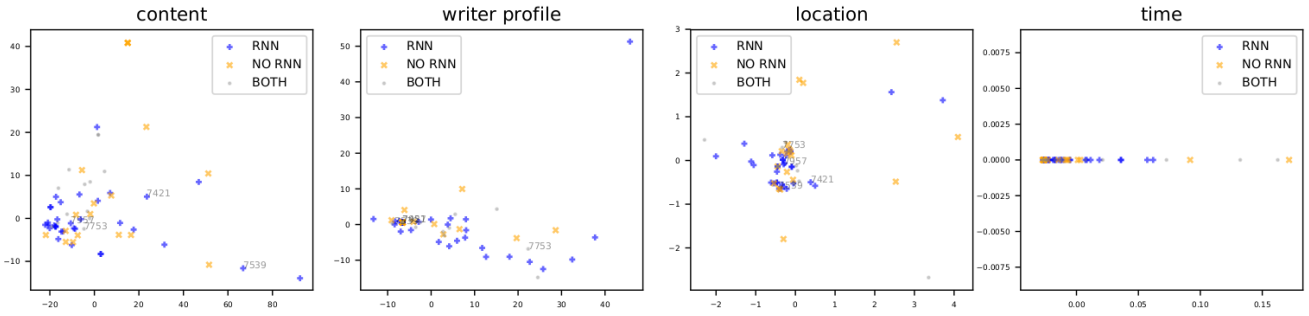


Figure (b): Relevant tweets found for event 100: "A mass drug overdose at a home in Chico, California, has killed one person ...". A center tweet 7753: "Good ole Mexican drugs in sanctuary Pelosi state of California". A non-center tweet 7421: "Love the people on here thinking a wall can stop Americas opioid addiction. Thats like hoping a rake can stop terrorism."

6. Searching for Microblogs Referring to Events by Deep Dynamic Query Strategies

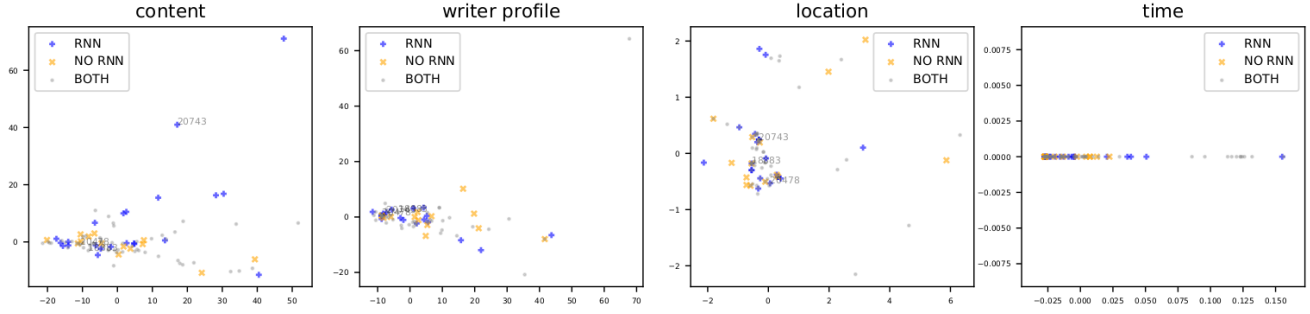


Figure (c): Relevant tweets found for event 180: "Multiple people have been shot at a SunTrust Bank in Sebring, Florida, authorities say ...". A center tweet 18883: "Really very sad news ... Now end the shooting ...". A non-center tweet 20743: "Build that wall."

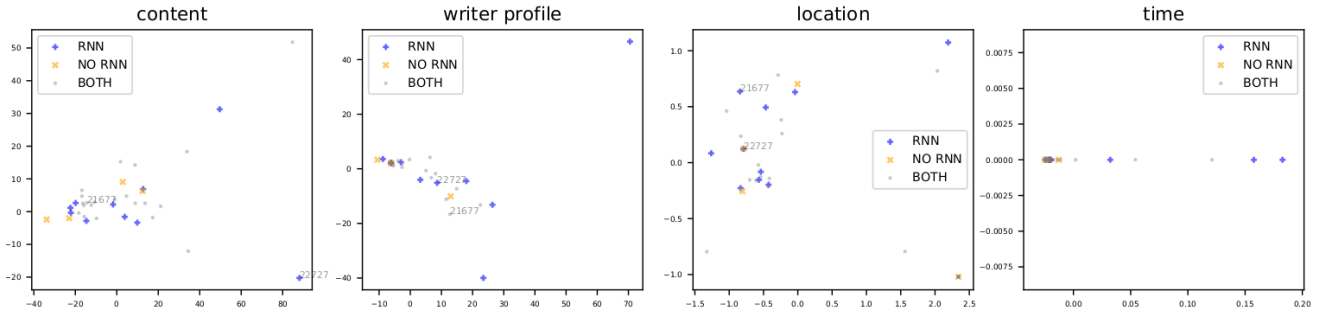


Figure (d): Relevant tweets found for event 209: "In the midst of a partial government shutdown stalemate over a border wall, the State Department cancels a conference ...". A center tweet 21677: "Wanna see the Democrats end the shutdown and start supporting the unpaid federal employees?". A non-center tweet 22727: "This needs to stop."

Figure 6.8: Visualization of relevant microblogs found by search processes applying with RNN model and without RNN model. We projected found microblogs into two dimensional spaces of textual contents, writer's profiles, locations, and times, respectively. Blue cross, orange x, and gray point are a microblog found by RNN, without RNN, and both of RNN and without RNN, respectively. In the content space of Figure 6.8 (b), for event 100: "A mass drug overdose at a home in Chico, California, has killed one person ...", both models could find representative microblogs, from the center of microblogs, such as tweet 7957: "Don't do drugs, simple!!"; while only RNN was able to find more diverse microblogs, which might be far from the center results, such as tweet 7421: "Love the people on here thinking a wall can stop Americas opioid addiction. Thats like hoping a rake can stop terrorism."

6. Searching for Microblogs Referring to Events by Deep Dynamic Query Strategies

Table 6.8: Performance of exploratory or exploitative query strategies.

Method	Recall	Implicit	Explicit
Ours (all query strategies)	0.561	0.331	0.229
Ours (only exploitation)	0.264	0.120	0.144
Ours (only exploration)	0.587	0.353	0.234

6.7.4 Effect of Exploratory Query Strategies

We evaluated the performance of the search process with different settings of query strategies. Table 6.8 showed that exploratory query strategies was important than the exploitative ones. Search by only exploratory query strategies could even achieved the same performance by applying all strategies (Ours). However, this indirectly showed that our search process could apply query strategies of exploration or exploitation well at the same time.

6.7.5 Robust to Ranking Criteria of Search Engine

We then check the performance of the search process in response to the changes of ranking criteria of search results, i.e., the changes of environment. Figure 6.9 showed the performance. The specific ranking criterion for some query feature could slightly affect our models (Ours), as shown that the performance was in the order of ranking favor for contents, time, locations, and writer profiles, respectively. However, our performance was kept when the ranking criterion was random and this might show the effect of our reinforcement learning. We answered **RQ4** that our models kept performance when the ranking criteria changed. Our search could work with both query strategies or with only exploratory query strategies.

6. Searching for Microblogs Referring to Events by Deep Dynamic Query Strategies

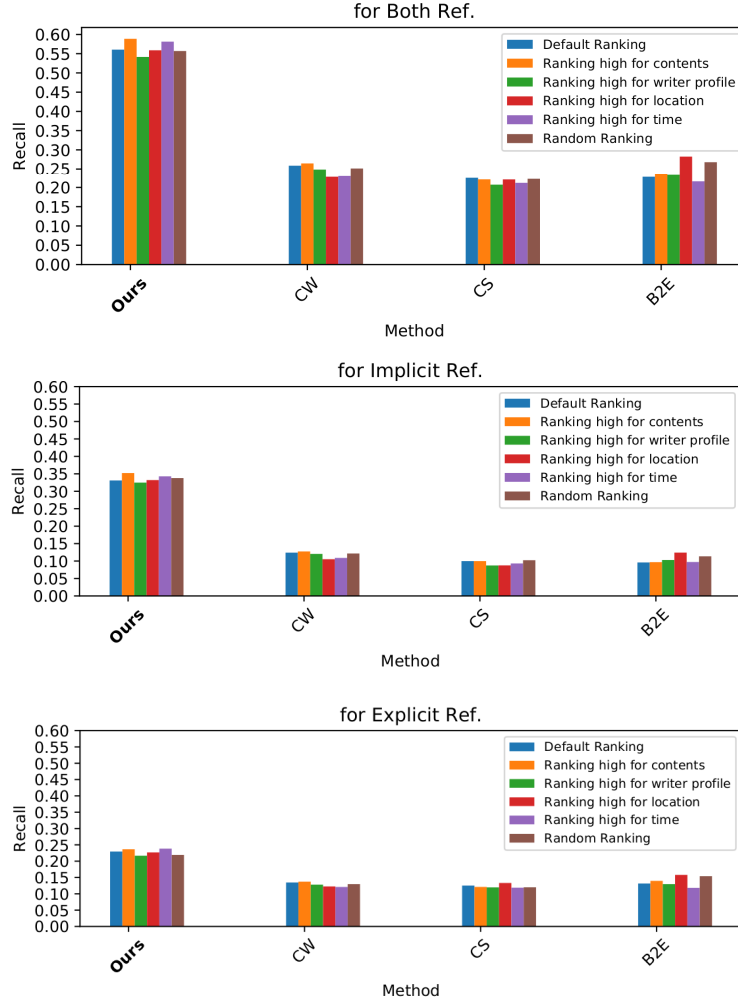


Figure 6.9: Microblog search performance under varying ranking criteria of search engine. The parameters used in ranking high for contents, writer profiles, locations, and time were $\alpha_c = 1.5$, $\alpha_u = 1.5$, $\alpha_l = 1.5$, and $\alpha_z = 1.5$, respectively. Note that the default ranking parameters were $\alpha_x = 1.0$.

6.7.6 Case Study

We conducted case study as shown in Table 6.9. Our methods found effective queries composed of non-keyword contents, such as sentiment words, cf. microblogs 1, 3, and 5. We also constructed effective queries for implicit references, such as those composed of related writer profiles or close locations, cf. microblogs 1, 2, and 4. However, the baseline method did not work. For example, corpus-based search did not find effective non-keyword queries. Ours and the baseline method could find queries for explicit reference, cf. microblogs 5 and 6.

6. Searching for Microblogs Referring to Events by Deep Dynamic Query Strategies

Table 6.9: Case study of searching for microblogs referring to the event, “Suspect confesses to kidnapping Jayme Closs and killing her parents, and says he decided to target her after seeing”. Note that we list queries used by a method that retrieved microblogs and “-” means no such queries. Exploitative query strategies for content, writer profile, post location, and post time are denoted by $a_{1,c}$, $a_{1,u}$, $a_{1,l}$, and $a_{1,z}$. Similarly, Exploratory query strategies for those items are denoted by $a_{2,c}$, $a_{2,u}$, $a_{2,l}$, and $a_{2,z}$.

	Microblog	Ref. type	CS	Ours (PS)	Ours
1	c:“Praying for this brave young woman.” u:“Mid-century woman with decades of music in my head ...” l:“Florida” z:“Jan 14 20:53:24 2019”	Imp.	-	-	strategy $a_{2,u}$ q{c:“pray” “brave” u:“woman”}
2	c:“Turn this sicko into a twice baked potato. Poor girl.” l:“Pittsburgh, PA” z:“Jan 14 20:50:33 2019”	Imp.	-	-	strategy $a_{2,l}$ q{c:“sicko” l:(close to Pitts- burgh)}
3	c:“Sad and uncivilised moved by one from the civilised domains” l:“Gombe, Nigeria” z:“Jan 15 11:05:09 2019”	Imp.	-	strategy $a_{2,c}$ q{c:“sad” “un- civilised”}	strategy $a_{2,c}$ q{c:“sad” “civilised”}
4	c:“I know they always pointe out crimes com- mitted by non-white and then tell us that they are not racist ...” u:“General News Politics History” l:“Bellingham, WA” z:“Jan 15 15:02:14 2019”	Imp.	q{c:“crime” “racist”}	strategy $a_{2,c}$ q{c:“crime” “racist”}	strategy $a_{1,u}$ q{c:“crime” u:“politics”}
5	c:“I wish this perv got killed in a shootout with cops! Poor little Jayme will now need to endure narrating ...” u:“UnCommented RT is an endorsement” z:“Jan 15 04:18:01 2019”	Exp.	q{c:“jayme” “kill”}	strategy $a_{1,c}$ q{c:“jayme”}	strategy $a_{2,c}$ q{c:“jayme” “poor”}
6	c:“Jayme you are a brave girl. He shall go to hell.” u:“daughter, sister, mother, lover, friend. scrapbook event organizer. loves to laugh. loves to live.” z:“Jan 14 22:31:42 2019”	Exp.	q{c:“jayme”}	strategy $a_{1,c}$ q{c:“jayme”}	strategy $a_{1,c}$ q{c:“jayme”}

6.8 Discussion

We listed the findings from the experimental results as follows.

- **Our method could search for microblogs implicitly or explicitly referring to events.** It was also applicable to unseen events.
- **We propose a robust reinforcement learning framework, with exploratory and exploitative query strategies.** Applying exploratory and exploitative query strategies was essential to search for implicit or explicit references, while patterns of query strategies did not work. Our search process outperformed baseline methods such as corpus-based and word-to-word-mapping based.
- **Optimizing online query strategies by estimating rewards was effective.** Ours was effective compared to searching without estimating rewards in offline policy, and relevance estimation of microblogs affected. I.e., search performance improved when estimating rewards was accurate.
- **RNN model with sequential search experiences could improve.** Searching by using only current search states had limited improvement.
- **Our dynamic search process was robust to ranking criteria of search engine.** Our model could keep performance in random ranking criteria of search engine.
- **Searching without analyzing search results or without feedback (rewards) might not work.** One example was that only applying relevant terms to the target event could not work for search. Furthermore, the quality of feedback (rewards) might affect as the third point above.

6. Searching for Microblogs Referring to Events by Deep Dynamic Query Strategies

We further discussed the limitation of this research or future work as follows.

- **Optimization of reinforcement learning framework for microblog search.** In this research, we proposed a framework based on deep Q-learning with RNN model. However, there are issues to be addressed. (a) Timing of optimizing online query strategies. In our or common modeling, the strategies were optimized with constant time steps. We wonder that there is a principle telling that one optimization should be taken because it will improve query strategies. (b) More appropriate query strategies or more features for search states. We might show the effectiveness of exploratory or exploitative strategies but are still not sure how to measure if query strategies or search states are appropriate or enough.

7. CONCLUSIONS

7.1 Summary

In this study, we discussed approaches for linking the implicit references in microblogs, to real-world knowledge (such as entities or events). For resolution or retrieval of implicit entity references, we proposed methodologies to complement the deficiency out of common NLP techniques. We linked opinions, expressions, or feelings in microblogs to entities or events by crossing the mismatch induced by heterogeneous properties of multiple media. We proposed methodologies upon approaches of deep learning, transfer learning, or reinforcement learning. The contributions in the four researches addressed in this study are listed as follows.

- (1) To identify multiple indirectly or directly referred entities in a microblog:
 - We proposed an accurate and computationally efficient CRF model with adaptive dependency. We also proposed effective features for indirect references.
- (2) To identify implicitly or explicitly referred events from microblogs:
 - We proposed deep transfer learning with dual transformations (DT).
 - We also proposed reference-type-based DT and applied DT with dense modeling, which was effective on implicit and explicit references and unseen events.
 - We proposed reference-type-based event contents generation, with mining similar links among microblogs and events.

7. Conclusions

(3) Searching for microblogs implicitly or explicitly referring to events, through search engine:

- We proposed exploratory and exploitative query strategies that could search for microblogs implicitly or explicitly referring to events.
- We proposed robust reinforcement learning with exploratory and exploitative query strategies. We predicted long-term returns with RNN model using sequential search states.
- We proposed a highly applicable search process that could be applied to unseen or new events. To this end, we optimized online query strategies by estimating rewards using Chapter 5.
- We provided our datasets and a reference implementation of our approaches¹¹.

Social contributions

- For the critical social issues arising in recent social media, such as fake news, misinformation, disinformation, and threats of cyber armies, researchers may want to find more clues from the real world or from people who concern for unevaluated textual information (e.g., people's posting) in social media sites (e.g., Twitter), our technology can be a tool that accesses the related real world information or the related information from people who concern in the social media site.
- For the survey of the public opinion, one effective way is to utilize online questionnaire but this way is professional to normal users and may take cost of time. As a trial and low-cost way, we may provide a tool that automatically extracts relevant people's posts in social media sites, which might contain opinions related to the survey. Especially, we may find diverse or useful information because people often share his/her opinions by writing posts in social media sites.

¹¹ https://github.com/JunliLu/search_ai

7.2 Future Directions

There are several valuable research topics related to this study. We list some possible research directions by considering related research domains and the powerful technologies of deep learning approaches as follows.

- Reinforcement learning (RL) approaches for search tasks, such as microblog search or web search, by considering more contexts in search behaviors or the domains of search. In this study, we tackled a microblog search problem for events by proposing a search agent using reinforcement learning. However, in real world, a search task can be more complicate by considering the factors of human searchers. For example, human searchers may have various intents for making a search task, and therefore the relevant search results can be defined differently or even changed during a search process. Moreover, human searchers may have various requirements for the searched results, and thus the evaluation of search results or a search process can be defined differently. In other words, for microblog/web search by human searchers, suppose we want to assist searchers by RL approaches, we may require to encode the complicated factors of human searchers into the corresponding elements of RL, such as state, action, and return, which is challenging because modeling the factors of human searchers can be much complicated.
- Transfer learning or representation learning approaches for connecting social media posting to real world information. In this study, we utilized the following features for the representations of microblogs or real world information (e.g., entities or events), which are, the links, syntactic patters, topic information, textual contents, writer's contents, posted locations, and time. However, there are more factors for those representations, which can be effective to link microblogs to the dynamics of real world. For example, the representations of microblogs from different regions or different groups of users should be treated differently since word sensing can be quite different across regions or groups of users. The representations for real world information are vice versa. Moreover, as microblogs might be posed with specific motivations or involved with the interactions among users, the posting behaviors of social media users can be effective clues for the representations. Meanwhile, as the dynamics of social

7. Conclusions

media or real world information changes or evolves all the times, the learned knowledge for the representations or the matching tasks, in the past, might be inappropriate for the novel information or in the future time. Therefore, the effective transfer learning is required to utilize the past gained knowledge into the dynamics among the big data of social media and real world information.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisor Professor Katsumi Tanaka. During my PhD study, Prof. Tanaka always supported my research with his rich research experiences.

I would like to express my sincere gratitude to professors of thesis committee, Professor Masatoshi Yoshikawa, Professor Keishi Tajima, and Professor Sadao Kurohashi. Each professor gave me valuable comments for the thesis.

I would like to express my sincere gratitude to Associate Professor Makoto P. Kato and Associate Professor Takehiro Yamamoto, who gave me valuable comments and discussed research idea with me.

I would like to express my sincere gratitude to Professor Kazutoshi Sumiya and Professor Ee-Peng Lim, who gave me valuable comments for research topics.

I would like to express my sincere gratitude to all people who helped me during my PhD study, from Tanaka Laboratory, Kyoto University, or research institutes or companies I worked in.

I would like to express my sincere gratitude to my family, who supported me during my PhD study, including my wife, parents-in-law, and my parents.

BIBLIOGRAPHY

- [1] Stilo, G. and Velardi, P.: Efficient temporal mining of micro-blog texts and its application to event discovery, *Data Min. Knowl. Discov.*, Vol. 30, No. 2, pp. 372–402 (2016).
- [2] Paltoglou, G.: Sentiment-based event detection in Twitter, *JASIST*, Vol. 67, No. 7, pp. 1576–1587 (2016).
- [3] Moran, S., McCreadie, R., Macdonald, C. and Ounis, I.: Enhancing First Story Detection using Word Embeddings, *SIGIR 2016*, pp. 821–824 (2016).
- [4] Ritter, A., Mausam, Etzioni, O. and Clark, S.: Open domain event extraction from twitter, *KDD 2012*, pp. 1104–1112 (2012).
- [5] Weng, J. and Lee, B.: Event Detection in Twitter, *ICWSM 2011* (2011).
- [6] Becker, H., Naaman, M. and Gravano, L.: Beyond Trending Topics: Real-World Event Identification on Twitter, *ICWSM 2011* (2011).
- [7] Abdelhaq, H., Sengstock, C. and Gertz, M.: EvenTweet: Online Localized Event Detection from Twitter, *PVLDB*, Vol. 6, No. 12, pp. 1326–1329 (2013).
- [8] Sakaki, T., Okazaki, M. and Matsuo, Y.: Earthquake shakes Twitter users: real-time event detection by social sensors, *WWW 2010*, pp. 851–860 (2010).
- [9] Wang, Q., She, J., Song, T., Tong, Y., Chen, L. and Xu, K.: Adjustable Time-Window-Based Event Detection on Twitter, *WAIM 2016*, pp. 265–278 (2016).
- [10] Jatowt, A., Antoine, É., Kawai, Y. and Akiyama, T.: Mapping Temporal Horizons: Analysis of Collective Future and Past related Attention in Twitter, *WWW 2015*, pp. 484–494 (2015).
- [11] Petrovic, S., Osborne, M. and Lavrenko, V.: Using paraphrases for improving first story detection in news and Twitter, *NAACL HLT 2012*, pp. 338–346 (2012).

- [12] Petrovic, S., Osborne, M. and Lavrenko, V.: Streaming First Story Detection with application to Twitter, NAACL HLT 2010, pp. 181–189 (2010).
- [13] Antoine, É., Jatowt, A., Wakamiya, S., Kawai, Y. and Akiyama, T.: Portraying Collective Spatial Attention in Twitter, KDD 2015, pp. 39–48 (2015).
- [14] Benhardus, J. and Kalita, J.: Streaming trend detection in Twitter, IJWBC, Vol. 9, No. 1, pp. 122–139 (2013).
- [15] Aiello, L. M., Petkos, G., Martín, C. J., Corney, D., Papadopoulos, S., Skraba, R., Göker, A., Kompatsiaris, I. and Jaimes, A.: Sensing Trending Topics in Twitter, IEEE Trans. Multimedia, Vol. 15, No. 6, pp. 1268–1282 (2013).
- [16] Nichols, J., Mahmud, J. and Drews, C.: Summarizing sporting events using twitter, IUI 2012, pp. 189–198 (2012).
- [17] Marcus, A., Bernstein, M. S., Badar, O., Karger, D. R., Madden, S. and Miller, R. C.: Twitinfo: aggregating and visualizing microblogs for event exploration, CHI 2011, pp. 227–236 (2011).
- [18] Chakrabarti, D. and Punera, K.: Event Summarization Using Tweets, ICWSM 2011 (2011).
- [19] Stefanidis, A., Crooks, A. and Radzikowski, J.: Harvesting ambient geospatial information from social media feeds, GeoJournal, Vol. 78, No. 2, pp. 319–338 (2013).
- [20] Ghiassi, M., Skinner, J. and Zimbra, D.: Twitter brand sentiment analysis: A hybrid system using n-gram analysis and dynamic artificial neural network, Expert Syst. Appl., Vol. 40, No. 16, pp. 6266–6282 (2013).
- [21] Conover, M., Gonçalves, B., Ratkiewicz, J., Flammini, A. and Menczer, F.: Predicting the Political Alignment of Twitter Users, SocialCom 2011, pp. 192–199 (2011).
- [22] Allan, J.(ed.): Topic Detection and Tracking: Event-based Information Organization, Kluwer Academic Publishers, Norwell, MA, USA (2002).
- [23] Petrovic, S., Osborne, M., McCreadie, R., Macdonald, C., Ounis, I. and Shrimpton, L.: Can Twitter Replace Newswire for Breaking News?, ICWSM 2013 (2013).
- [24] Zhao, W. X., Jiang, J., Weng, J., He, J., Lim, E., Yan, H. and Li, X.: Comparing Twitter and Traditional Media Using Topic Models, ECIR 2011, pp. 338–349 (2011).
- [25] Watanabe, K., Ochi, M., Okabe, M. and Onai, R.: Jasmine: a real-time local-event detection system based on geolocation information propagated to microblogs, CIKM 2011, pp. 2541–2544 (2011).
- [26] Hua, T., Chen, F., Zhao, L., Lu, C. and Ramakrishnan, N.: Automatic targeted-domain spatiotemporal event detection in twitter, GeoInformatica, Vol. 20, No. 4, pp. 765–795 (2016).

- [27] Zhang, Y. and Qu, Z.: A novel method for online bursty event detection on Twitter, ICSESS 2015, pp. 284–288 (2015).
- [28] Zhou, X. and Chen, L.: Event detection over twitter social media streams, VLDB J., Vol. 23, No. 3, pp. 381–400 (2014).
- [29] Sankaranarayanan, J., Samet, H., Teitler, B. E., Lieberman, M. D. and Sperling, J.: TwitterStand: news in tweets, SIGSPATIAL 2009, pp. 42–51 (2009).
- [30] Domeniconi, G., Moro, G., Pasolini, R. and Sartori, C.: A Study on Term Weighting for Text Categorization: A Novel Supervised Variant of tf.idf, DATA 2015, pp. 26–37 (2015).
- [31] Jain, A. K.: Data clustering: 50 years beyond K-means, Pattern Recognition Letters, Vol. 31, No. 8, pp. 651–666 (2010).
- [32] Shen, W., Wang, J. and Han, J.: Entity Linking with a Knowledge Base: Issues, Techniques, and Solutions, IEEE Trans. Knowl. Data Eng., Vol. 27, No. 2, pp. 443–460 (2015).
- [33] Lu, J., Kato, M. P., Yamamoto, T. and Tanaka, K.: Entity Identification on Microblogs by CRF Model with Adaptive Dependency, IEICE Transactions, Vol. 99-D, No. 9, pp. 2295–2305 (2016).
- [34] Joachims, T., Finley, T. and Yu, C. J.: Cutting-plane training of structural SVMs, Machine Learning, pp. 27–59 (2009).
- [35] Vedaldi, A.: A MATLAB wrapper of SVM^{struct} , <http://www.robots.ox.ac.uk/vedaldi/svmstruct.html> (2011).
- [36] W. Shen, J. Wang, and J. Han, “Entity linking with a knowledge base: Issues, techniques, and solutions,” IEEE TKDE, pp. 443–460, 2015.
- [37] W. Shen, J. Han, and J. Wang, “A probabilistic model for linking named entities in web text with heterogeneous information networks,” in SIGMOD, 2014, pp. 1199–1210.
- [38] C. A. Sutton and A. McCallum, “An introduction to conditional random fields,” Foundations and Trends in Machine Learning, pp. 267–373, 2012.
- [39] A. McCallum, “Efficiently inducing features of conditional random fields,” CoRR, vol. abs/1212.2504, 2012.
- [40] L. Derczynski, D. Maynard, G. Rizzo, M. van Erp, G. Gorrell, R. Troncy, J. Petrak, and K. Bontcheva, “Analysis of named entity recognition and linking for tweets,” Inf. Process. Manage., pp. 32–49, 2015.
- [41] C. Li, A. Sun, J. Weng, and Q. He, “Tweet segmentation and its application to named entity recognition,” IEEE TKDE, pp. 558–570, 2015.
- [42] S. Guo, M. Chang, and E. Kiciman, “To link or not to link? A study on end-to-end tweet entity linking,” in NAACL HLT, 2013, pp. 1020–1030.

- [43] X. Liu, F. Wei, S. Zhang, and M. Zhou, “Named entity recognition for tweets,” *ACM TIST*, p. 3, 2013.
- [44] X. Liu and M. Zhou, “Two-stage NER for tweets with clustering,” *Inf. Process. Manage.*, pp. 264–273, 2013.
- [45] J. Hoffart, Y. Altun, and G. Weikum, “Discovering emerging entities with ambiguous names,” in *WWW*, 2014, pp. 385–396.
- [46] Z. Guo and D. Barbosa, “Robust entity linking via random walks,” in *CIKM*, 2014, pp. 499–508.
- [47] W. Shen, J. Wang, P. Luo, and M. Wang, “Linking named entities in tweets with knowledge base via user interest modeling,” in *KDD*, 2013, pp. 68–76.
- [48] Y. Li, C. Wang, F. Han, J. Han, D. Roth, and X. Yan, “Mining evidences for named entity disambiguation,” in *KDD*, 2013, pp. 1070–1078.
- [49] R. Blanco, G. Ottaviano, and E. Meij, “Fast and space-efficient entity linking for queries,” in *WSDM*, 2015, pp. 179–188.
- [50] C. Wang, K. Chakrabarti, T. Cheng, and S. Chaudhuri, “Targeted disambiguation of ad-hoc, homogeneous sets of named entities,” in *WWW*, 2012, pp. 719–728.
- [51] E. Meij, W. Weerkamp, and M. de Rijke, “Adding semantics to microblog posts,” in *WSDM*, 2012, pp. 563–572.
- [52] B. Skaggs and L. Getoor, “Topic modeling for wikipedia link disambiguation,” *ACM Trans. Inf. Syst.*, p. 10, 2014.
- [53] X. Han and L. Sun, “An entity-topic model for entity linking,” in *EMNLP-CoNLL*, 2012, pp. 105–115.
- [54] D. Milne and I. H. Witten, “An effective, low-cost measure of semantic relatedness obtained from wikipedia links,” in *AAAI*, 2008, pp. 25–30.
- [55] M. P. Kato, H. Ohshima, S. Oyama, and K. Tanaka, “Query by analogical example: relational search using web search engine indices,” in *CIKM*, 2009, pp. 27–36.
- [56] E. Shutova, S. Teufel, and A. Korhonen, “Statistical metaphor processing,” *Computational Linguistics*, pp. 301–353, 2013.
- [57] E. Shutova, B. Devereux, and A. Korhonen, “Conceptual metaphor theory meets the data: a corpus-based human annotation study,” *Language Resources and Evaluation*, pp. 1261–1284, 2013.
- [58] T. Pedersen, S. Patwardhan, and J. Michelizzi, “Wordnet: : Similarity - measuring the relatedness of concepts,” in *AAAI*, 2004, pp. 1024–1025.
- [59] T. Finley and T. Joachims, “Training structural svms when exact inference is intractable,” in *ICML*, 2008, pp. 304–311.

- [60] T. Joachims, T. Finley, and C. J. Yu, “Cutting-plane training of structural svms,” *Machine Learning*, pp. 27–59, 2009.
- [61] E. B. Sudderth, A. T. Ihler, M. Isard, W. T. Freeman, and A. S. Willsky, “Nonparametric belief propagation,” *Commun. ACM*, pp. 95–103, 2010.
- [62] A. Vedaldi, “A MATLAB wrapper of SVM struct ,” <http://www.vlfeat.org/~vedaldi/code/svm-struct-matlab.html>, 2011.
- [63] S. Chatterjee and S. J. Russell, “A temporally abstracted viterbi algorithm,” *CoRR*, vol. abs/1202.3707, 2012.
- [64] Farag, M. M. G., Lee, S. and Fox, E. A.: Focused crawler for events, *Int. J. on Digital Libraries*, Vol. 19, No. 1, pp. 3–19 (2018).
- [65] Klein, M., Balakireva, L. and de Sompel, H. V.: Focused Crawl of Web Archives to Build Event Collections, in *WebSci*, pp. 333–342 (2018).
- [66] Gottschalk, S. and Demidova, E.: EventKG: A Multilingual Event-Centric Temporal Knowledge Graph, in *ESWC*, pp. 272–287 (2018).
- [67] Gossen, G., Demidova, E. and Risse, T.: Extracting Event-Centric Document Collections from Large-Scale Web Archives, in *TPDL*, pp. 116–127 (2017).
- [68] Farag, M. M. G.: Intelligent Event Focused Crawling, PhD Thesis, Virginia Tech, Blacksburg, VA, USA (2016).
- [69] Zhang, C., Lei, D., Yuan, Q., Zhuang, H., Kaplan, L. M., Wang, S. and Han, J.: GeoBurst+: Effective and Real-Time Local Event Detection in Geo-Tagged Tweet Streams, *ACM TIST*, Vol. 9, No. 3, pp. 34:1–34:24 (2018).
- [70] Zhang, C., Liu, L., Lei, D., Yuan, Q., Zhuang, H., Hanratty, T. and Han, J.: TrioVecEvent: Embedding-Based Online Local Event Detection in Geo-Tagged Tweet Streams, in *SIGKDD*, pp. 595–604 (2017).
- [71] Han, M., Wuillemin, P. and Senellart, P.: Focused Crawling Through Reinforcement Learning, in *ICWE*, pp. 261–278 (2018).
- [72] Xia, L., Xu, J., Lan, Y., Guo, J., Zeng, W. and Cheng, X.: Adapting Markov Decision Process for Search Result Diversification, in *SIGIR*, pp. 535–544 (2017).
- [73] Odijk, D., Meij, E., Sijaranamual, I. and de Rijke, M.: Dynamic Query Modeling for Related Content Finding, in *SIGIR*, pp. 33–42 (2015).
- [74] Jiang, L., Wu, Z., Feng, Q., Liu, J. and Zheng, Q.: Efficient Deep Web Crawling Using Reinforcement Learning, in *PAKDD*, pp. 428–439 (2010).
- [75] Brittain, M., Bertram, J. R., Yang, X. and Wei, P.: Prioritized Sequence Experience Replay, *CoRR*, Vol. abs/1905.12726 (2019).

- [76] van Hasselt, H., Guez, A. and Silver, D.: Deep Reinforcement Learning with Double Q-Learning, in AAAI, pp. 2094–2100 (2016).
- [77] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D. and Riedmiller, M. A.: Playing Atari with Deep Reinforcement Learning, CoRR, Vol. abs/1312.5602 (2013).
- [78] Xian, Y., Lampert, C. H., Schiele, B. and Akata, Z.: Zero-Shot Learning - A Comprehensive Evaluation of the Good, the Bad and the Ugly, IEEE Trans. Pattern Anal. Mach. Intell., Vol. 41, No. 9, pp. 2251–2265 (2019).
- [79] Goodfellow, I., Bengio, Y. and Courville, A.: Deep Learning, MIT Press.
- [80] Li, X., Guo, Y. and Schuurmans, D.: Semi-Supervised Zero-Shot Classification with Label Representation Learning, in ICCV, pp. 4211–4219 (2015).
- [81] Lampert, C. H., Nickisch, H. and Harmeling, S.: Attribute-Based Classification for Zero-Shot Visual Object Categorization, IEEE Trans. Pattern Anal. Mach. Intell., Vol. 36, No. 3, pp. 453–465 (2014).
- [82] Rohrbach, M., Stark, M. and Schiele, B.: Evaluating knowledge transfer and zero-shot learning in a large-scale setting, in CVPR, pp. 1641–1648 (2011).
- [83] Rohrbach, M., Stark, M., Szarvas, G., Gurevych, I. and Schiele, B.: What helps where - and why? Semantic relatedness for knowledge transfer, in CVPR, pp. 910–917 (2010).
- [84] Zhang, Z. and Saligrama, V.: Zero-Shot Learning via Joint Latent Similarity Embedding, in CVPR, pp. 6034–6042 (2016).
- [85] Changpinyo, S., Chao, W., Gong, B. and Sha, F.: Synthesized Classifiers for Zero-Shot Learning, in CVPR, pp. 5327–5336 (2016).
- [86] Akata, Z., Reed, S. E., Walter, D., Lee, H. and Schiele, B.: Evaluation of output embeddings for fine-grained image classification, in CVPR, pp. 2927–2936 (2015).
- [87] Wang, X., Ye, Y. and Gupta, A.: Zero-Shot Recognition via Semantic Embeddings and Knowledge Graphs, in CVPR, pp. 6857–6866 (2018).
- [88] Xian, Y., Akata, Z., Sharma, G., Nguyen, Q. N., Hein, M. and Schiele, B.: Latent Embeddings for Zero-Shot Classification, in CVPR, pp. 69–77 (2016).
- [89] Frome, A., Corrado, G. S., Shlens, J., Bengio, S., Dean, J., Ranzato, M. and Mikolov, T.: DeViSE: A Deep Visual-Semantic Embedding Model, in NIPS, pp. 2121–2129 (2013).
- [90] Zacks, J. M. and Tversky, B.: Event structure in perception and conception, Psychological bulletin, Vol. 127, No. 1, pp. 3–21 (2001).
- [91] Liu, B., Kato, M. P. and Tanaka, K.: Cognition-Aware Summarization of Photos Representing Events, IEICE Transactions, Vol. 99-D, No. 12, pp. 3140–3153 (2016).

- [92] François-Lavet, V., Henderson, P., Islam, R., Bellemare, M. G. and Pineau, J.: An Introduction to Deep Reinforcement Learning, *Foundations and Trends in Machine Learning*, Vol. 11, No. 3-4, pp. 219–354 (2018).
- [93] Gal, Y. and Ghahramani, Z.: A Theoretically Grounded Application of Dropout in Recurrent Neural Networks, in *NIPS*, pp. 1019–1027 (2016).
- [94] Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R. and Schmidhuber, J.: LSTM: A Search Space Odyssey, *IEEE Trans. Neural Netw. Learning Syst.*, Vol. 28, No. 10, pp. 2222–2232 (2017).
- [95] Olah, C.: Understanding LSTM Networks (2015).
- [96] Agarap, A. F.: Deep Learning using Rectified Linear Units (ReLU), *CoRR*, Vol. abs/1803.08375 (online), available from <http://arxiv.org/abs/1803.08375> (2018).
- [97] Zamani, H. and Croft, W. B.: Relevance-based Word Embedding, in *SIGIR*, pp. 505–514 (2017).
- [98] Mitra, B. and Craswell, N.: Neural Text Embeddings for Information Retrieval, in *WSDM*, pp. 813–814 (2017).
- [99] Roy, D., Paul, D., Mitra, M. and Garain, U.: Using Word Embeddings for Automatic Query Expansion, *CoRR*, Vol. abs/1606.07608 (2016).
- [100] Diaz, F., Mitra, B. and Craswell, N.: Query Expansion with Locally-Trained Word Embeddings, in *ACL* (2016).
- [101] Bendersky, M., Pueyo, L. G., Harmsen, J. J., Josifovski, V. and Lepikhin, D.: Up next: retrieval methods for large scale related video suggestion, in *KDD*, pp. 1769–1778 (2014).
- [102] Chen, W., Cai, F., Chen, H. and de Rijke, M.: Attention-based Hierarchical Neural Query Suggestion, in *SIGIR*, pp. 1093–1096 (2018).
- [103] Song, J., Xiao, J., Wu, F., Wu, H., Zhang, T., Zhang, Z. M. and Zhu, W.: Hierarchical Contextual Attention Recurrent Neural Network for Map Query Suggestion, *IEEE Trans. Knowl. Data Eng.*, Vol. 29, No. 9, pp. 1888–1901 (2017).
- [104] Sordoni, A., Bengio, Y., Vahabi, H., Lioma, C., Simonsen, J. G. and Nie, J.: A Hierarchical Recurrent Encoder-Decoder for Generative Context-Aware Query Suggestion, in *CIKM*, pp. 553–562 (2015).
- [105] Chiu, C., Sainath, T. N., Wu, Y., Prabhavalkar, R., Nguyen, P., Chen, Z., Kannan, A., Weiss, R. J., Rao, K., Gonina, E., Jaitly, N., Li, B., Chorowski, J. and Bacchiani, M.: State-of-the-Art Speech Recognition with Sequence-to-Sequence Models, in *ICASSP*, pp. 4774–4778 (2018).

- [106] Nallapati, R., Zhou, B., dos Santos, C. N., Gülçehre, Ç. And Xiang, B.: Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond, in CoNLL, pp. 280–290 (2016).
- [107] Finkel, J. R., Grenager, T. and Manning, C. D.: Incorporating Non-local Information into Information Extraction Systemsby Gibbs Sampling, in ACL, pp. 363–370 (2005).
- [108] Wu, F., Liu, J., Wu, C., Huang, Y. and Xie, X.: NeuralChinese Named Entity Recognition via CNN-LSTM-CRF and Joint Training with Word Segmentation, in WWW, pp. 3342–3348 (2019).
- [109] Dong, C., Zhang, J., Zong, C., Hattori, M. and Di, H.: Character-Based LSTM-CRF with Radical-Level Features for Chinese Named Entity Recognition, in NLPCC, pp. 239–250 (2016).
- [110] Zubiaga, A.: A longitudinal assessment of the persistence of twitter datasets, in JASIST, Vol. 69, No. 8, pp. 974–984(2018).
- [111] Burel, G., Saif, H. and Alani, H.: Semantic Wide and Deep Learning for Detecting Crisis-Information Categories on Social Media, in ISWC, pp. 138–155 (2017).
- [112] Burel, G., Saif, H., Fernandez, M. and Alani, H.: On Semantics and Deep Learning for Event Detection in Crisis Situations, in ESWC (2017).
- [113] Burel, G. and Alani, H.: Crisis Event Extraction Service(CREES) – Automatic Detection and Classification of Crisis-related Content on Social Media, in ISCRAM (2018).
- [114] Nguyen, V. Q., Yang, H., Kim, K. and Oh, A.: Real-Time Earthquake Detection Using Convolutional Neural Network and Social Data, in BigMM, pp. 154–157 (2017).
- [115] Jain, P., Ross, R. and Schoen-Phelan, B.: Estimating Distributed Representation Performance in Disaster-Related Social Media Classification (2019).
- [116] Hausknecht, M. J. and Stone, P.: Deep Recurrent Q-Learning for Partially Observable MDPs, in AAAI, pp. 29–37 (2015).
- [117] Du, Y., Liu, W., Lv, X. and Peng, G.: An improved focused crawler based on Semantic Similarity Vector Space Model, Appl. Soft Comput., Vol. 36, pp. 392–407 (2015).
- [118] Xu, G., Jiang, P., Ma, C., Daneshmand, M. and Xie, S.: VRPSOFC: a framework for focused crawler using mutation improving particle swarm optimization algorithm, in TURC, pp. 4:1–4:7 (2019).
- [119] Seyfi, A. and Patel, A.: A focused crawler combinatory link and content model based on T-Graph principles, Computer Standards & Interfaces, Vol. 43, pp. 1–11 (2016).

PUBLICATIONS

Journal Papers

1. Jun-Li Lu, Makoto P. Kato, Takehiro Yamamoto, Katsumi Tanaka
“Entity Identification on Microblogs by CRF Model with Adaptive Dependency”,
IEICE Transactions on Information and Systems, Vol. 99-D, No. 9, pp. 2295-2305, Sep. 2016.
2. Jun-Li Lu, Makoto P. Kato, Takehiro Yamamoto, Katsumi Tanaka
“Event Identification for Explicit and Implicit References on Microblogs”,
IPSJ Transactions on Databases (TOD), Vol. 25, pp. 505-513, Jan. 2017.
3. Jun-Li Lu, Makoto P. Kato, Takehiro Yamamoto, Katsumi Tanaka
“Searching for Microblogs Referring to Events by Deep Dynamic Query Strategies”, IPSJ Transactions on Databases (TOD), published from April 2020.

International Conference Papers

4. Jun-Li Lu, Makoto P. Kato, Takehiro Yamamoto, Katsumi Tanaka
“Entity Identification on Microblogs by CRF Model with Adaptive Dependency”,
Proc. of 2015 IEEE/WIC/ACM International Conference on Web Intelligence (WI 2015), pp. 333-340, 2015 (“**Best Student Paper Award**”).

5. Jun-Li Lu, Ling-Yin Wei, Mi-Yen Yeh, "Influence maximization in a social network in the presence of multiple influences and acceptances", 2014 International Conference on Data Science and Advanced Analytics (DSAA 2014), pp. 230-236, 2014.
6. Jun-Li Lu, Mi-Yen Yeh, Yu-Ching Hsu, Shun-Neng Yang, Chai-Hien Gan, Ming-Syan Chen, "Operating Electric Taxi Fleets: a New Dispatching Strategy with Charging Plans", 2012 IEEE International Electric Vehicle Conference (IEVC 2012), pp. 1-8, 2012.

Patent

7. Jun-Li Lu, Mi-Yen Yeh, Yu-Ching Hsu, Shun-Neng Yang, Chai-Hien Gan, Ming-Syan Chen, "Method for dispatching electric automobile with charging plan and system thereof", (Taiwan) R.O.C. patent number: I489401, International Patent Classification (IPC), Sep. 2013.

Workshop

8. Jun-Li Lu, Makoto P. Kato, Takehiro Yamamoto, Katsumi Tanaka "Toward Entity Identification of Relation, Metonymy, and Metaphor on Microblogs", The 6th International Workshop with Mentors on Databases, Web and Information Management for Young Researchers, 2014.