Deep Learning for Entity Analysis

A thesis submitted in partial fulfillment for the
degree of Doctor of Philosophy
in
Computer Science

at the
School of Natural and Computational Sciences,
Massey University, Albany,
New Zealand.

Feng Hou

February 2021

Abstract

Our research focuses on three sub-tasks of entity analysis: fine-grained entity typing (FGET), entity linking and entity coreference resolution. We aim at improving FGET and entity linking by exploiting the document-level type constraints and improving entity linking and coreference resolution by embedding fine-grained entity type information.

To extract more efficient feature representations and offset label noises in the datasets for FGET, we propose three transfer learning schemes: (i) transferring sub-word embeddings to generate more efficient out-of-vocabulary (OOV) embeddings for mentions; (ii) using a pre-trained language model to generate more efficient context features; (iii) using a pre-trained topic model to transfer the topic-type relatedness through topic anchors and select confusing fine-grained types at inference time. The pre-trained topic model can offset the label noises without retreating to coarse-grained types.

To reduce the distinctiveness of existing entity embeddings and facilitate the learning of contextual commonality for entity linking, we propose a simple yet effective method, FGS2EE, to inject fine-grained semantic information into entity embeddings. FGS2EE first uses the embeddings of semantic type words to generate semantic entity embeddings, and then combines them with existing entity embeddings through linear aggregation. Based on our entity embeddings, we have achieved new state-of-the-art performance on two of the five out-domain test sets for entity linking.

Further, we propose a method, DOC-AET, to exploit DOCument-level coherence of named entity mentions and anonymous entity type (AET) words/mentions. We learn embeddings of AET words from the AET words' inter-paragraph co-occurrence matrix. Then, we build AET entity embeddings and document AET context embeddings using the AET word embeddings. The AET coherence are computed using the AET entity embeddings and document context embeddings. By incorporating such coherence scores, DOC-AET has achieved new state-of-the-art results on three of the five out-domain test sets for entity linking.

We also propose LASE (Less Anisotropic Span Embeddings) schemes for coreference resolution. We investigate the effectiveness of these schemes with extensive experiments. Our ablation studies also provide valuable insights about the contextualized representations.

In summary, this thesis proposes four deep learning approaches for entity analysis. Extensive experiments show that we have achieved state-of-the-art performance on the three sub-tasks of entity analysis.

Acknowledgements

Working towards a Ph.D was never easy, but it was always rewarding and joyful. Not only it is about the research outputs, but also it is about the process of working with wonderful persons. This dissertation would not have been possible without the support of many people.

First and foremost, I would like to express my sincere gratitude to my supervisors and role models. To Professor Ruili Wang, my supervisor, you provided me with perfectly balanced guidance and freedom. You have shaped my thoughts, always critiqued my work in a constructive way, and helped me gain confidence in myself. You have been so generous with your time, reading and commenting on so many of my writings. You caught imprecise or unclear expressions, and helped me improve my writing skills. Your comments were always to the point! For all your guidance, encouragement and support: thank you! To Professor Yi Zhou, my co-supervisor, I am grateful for the invaluable insights and your concrete help in many aspects. Thank you for the insightful and fruitful feedback and comments.

I am thankful to many faculty members at the School of Natural and Computational Sciences. They provided valuable guidance and support through my doctoral research. To Dr Kristin Stock, thank you for your meticulous and professional suggestions on this thesis.

I am also thankful to my friends and my colleagues in Professor Wang's research group for their friendship, encouragement and valuable suggestions.

To my sister, Yanxia, who shouldered so much responsibilities for the care of our parents. To my parents for their love, support and upbringing which made me who I am today. To Fen, my wife, for always believing in me and for her unconditional love and support. And last but not least, to our son, Xuancheng, whose smile and funny jokes put everything in perspective and always relieve the stress.

Publications

The following research papers have been published in or submitted to International Journals and Conferences:

- **Feng Hou**, Ruili Wang, Jun He and Yi Zhou. 2020. Improving Entity Linking through Semantic Reinforced Entity Embeddings. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 6843 - 6848, July 5 - 10, 2020. Association for Computational Linguistics. CORE rank A*.
- **Feng Hou**, Ruili Wang and Yi Zhou. Transfer Learning for Fine-grained Entity Typing. Accepted by *Knowledge and Information Systems*, Springer. ERA rank B.
- **Feng Hou**, Ruili Wang, Steven Cahan, Lily Chen and Yi Zhou. 2020. Improving Entity Linking through Anonymous Entity Mentions. Under phase II review of AAAI 2021, AAAI. CORE rank A*.
- **Feng Hou**, Ruili Wang, Steven Cahan, Lily Chen, Xiaoyun Jia and Yi Zhou. Anisotropic Span Embeddings and the Negative Impact of Higher-order Inference for Coreference Resolution. Submitted to 2021 annual conference of ACL.
- Ruili Wang, **Feng Hou**, Steven Cahan, Lily Chen, Xiaoyun Jia and Wanting Ji. Fine-Grained Entity Typing with a Type Taxonomy: a Systematic Review. Submitted to *IEEE Transactions on Artificial Intelligence*.

Contents

Abstract iii

Acknowledgements iv

Publications v

List of Figures xi

List of Tables xiii

Abbreviations xv

1 Introduction 1
- 1.1 Introduction to Entity Analysis . . . 1
- 1.2 Motivation . . . 3
- 1.3 Research Objectives and Hypotheses . . . 5
- 1.4 Contributions . . . 8
- 1.5 Thesis Outline . . . 9

2 Transfer Learning for Fine-grained Entity Typing 13
- 2.1 Introduction . . . 14
- 2.2 Related Work . . . 16
  - 2.2.1 Fine-Grained Entity Typing . . . 16
  - 2.2.2 Transfer Learning in NLP . . . 17
    - 2.2.2.1 Transfer Learning Through Pre-trained Language Models . . . 17
    - 2.2.2.2 Transfer Learning Through Topic Models . . . 18
  - 2.2.3 Vector Representation of Words . . . 18
- 2.3 Mention-level FGET Task Definition . . . 18
- 2.4 Topic-Type Relatedness Hypothesis . . . 20
  - 2.4.1 Topic-Type Relatedness . . . 20
  - 2.4.2 Topic Anchor . . . 22
- 2.5 Approach . . . 23
  - 2.5.1 Typing Model . . . 23
  - 2.5.2 Transfer Learning for Feature Representations . . . 25
    - 2.5.2.1 Transfer Learning for Mention Embedding . . . 25
    - 2.5.2.2 Transfer Learning for Context Embedding . . . 26
  - 2.5.3 Transfer Learning Through Topic Model . . . 26

2.5.3.1 HMM-LDA Model . . . 27
2.5.3.2 Capturing Topic-Type Relatedness with HMM-LDA . . . 27
2.5.3.3 Topic Distribution Estimation . . . 28
2.5.3.4 Type Selection . . . 29
2.6 Experiments . . . 29
2.6.1 Datasets . . . 29
2.6.2 Preprocessing . . . 30
2.6.3 Baselines . . . 31
2.6.4 Experimental Setup . . . 32
2.6.4.1 Hyperparameter Settings . . . 32
2.6.4.2 Evaluation Metrics . . . 33
2.6.5 Results Comparison and Analysis . . . 34
2.6.5.1 Performance Analysis . . . 34
2.6.5.2 Error Analysis . . . 35
2.7 Conclusion and Future Work . . . 36

3 Improving Entity Linking through Semantic Reinforced Entity Embeddings 45
3.1 Introduction . . . 45
3.2 Entity Linking Background . . . 47
3.2.1 Task Description . . . 47
3.2.2 Local Models for Entity Linking . . . 48
3.2.3 Global Models for Entity Linking . . . 49
3.3 Related Work . . . 51
3.4 Motivation . . . 53
3.5 Extracting Fine-grained Semantic Types . . . 53
3.5.1 Semantic Type Dictionary . . . 54
3.5.2 Extracting Semantic Types . . . 54
3.5.3 Remapping Semantic Words . . . 54
3.6 FGS2EE: Injecting Fine-Grained Semantic Information into Entity Embeddings 55
3.6.1 Semantic Entity Embeddings . . . 55
3.6.2 Semantic Reinforced Entity Embeddings . . . 55
3.7 Experiments . . . 56
3.7.1 Datasets . . . 56
3.7.2 Evaluation Metrics . . . 56
3.7.3 Experimental Settings . . . 58
3.7.4 Results . . . 58
3.8 Conclusion . . . 60

4 Improving Entity Linking through Anonymous Entity Mentions 67
4.1 Introduction . . . 67
4.2 Background . . . 70
4.2.1 Named Entity Linking . . . 70
4.2.2 Local Score for Candidate Ranking . . . 70
4.2.3 Global Score for Candidate Ranking . . . 70
4.3 Related Work . . . 71
4.3.1 NEL Using Entity Type Information . . . 71
4.3.2 Word Embeddings . . . 72
4.3.3 Embeddings Aggregation . . . 72

4.3.4 Fine-grained Entity Typing . . . 72
4.4 DOC-AET: Method Overview . . . 73
4.4.1 Motivation . . . 73
4.4.2 AET Dictionary . . . 74
4.4.3 Process of DOC-AET Method . . . 74
4.5 Generate AET Word Embeddings . . . 74
4.5.1 Document-level Inter-paragraph Co-occurrence of AET Words . . . 75
4.5.2 Learn AET Word Embeddings . . . 76
4.6 Incorporating AET Scores . . . 76
4.6.1 Entity Embeddings from AET Words . . . 76
4.6.2 Document Context Embeddings from AET Words . . . 76
4.6.3 Local AET Scores Using Document Context . . . 77
4.6.4 Model Training . . . 77
4.7 Experiments . . . 78
4.7.1 Datasets for AET Word Embeddings . . . 78
4.7.2 Datasets for NEL . . . 78
4.7.3 Evaluation Metrics and Baselines . . . 78
4.7.4 Experimental Settings . . . 79
4.7.5 Results . . . 79
4.7.6 Ablation Analysis . . . 81
4.7.7 Model Complexity . . . 81
4.7.8 AET Word Embeddings Evaluation . . . 82
4.8 Conclusion . . . 83

5 Exploiting Less Anisotropic Span Representations for Entity Coreference Resolution 89
5.1 Introduction . . . 89
5.2 Background of End-to-End Neural Coreference Resolution . . . 91
5.2.1 Task Description . . . 91
5.2.2 First-order Coreference Resolution . . . 91
5.2.3 Higher-order Coreference Resolution . . . 92
5.2.4 Span Embeddings Based on Contextualized Representations . . . 93
5.3 Related Work . . . 94
5.4 Gauging Contextualized Representations . . . 94
5.5 Sources of Anisotropic Span Embeddings . . . 95
5.6 Generating Less Anisotropic Span Embeddings . . . 96
5.6.1 Lower Depth for Higher-order Refinement . . . 96
5.6.2 Using Penultimate Layer Embeddings . . . 96
5.6.3 Using Linear Aggregations of Multiple Layers Embeddings . . . 96
5.7 Experiments . . . 97
5.7.1 Implementation and Hyperparameters . . . 97
5.7.2 Baselines . . . 98
5.7.3 Data Sets and Evaluation Metrics . . . 98
5.7.3.1 Document Level Coreference Resolution: OntoNotes . . . 98
5.7.3.2 Paragraph Level Coreference Resolution: GAP . . . 99
5.7.3.3 Data Sets for Gauging Contextualized Representations . . . 100
5.7.4 Results and Findings . . . 100

5.7.4.1 Results 101
5.7.4.2 Findings and Analysis 103
5.8 Conclusion 105

6 Conclusion **111**
6.1 Research Summary 111
6.2 Future Directions 113

A Statement of Contribution **115**

List of Figures

1.1 Entity mentions in a sentence are coloured. Different background colors denote different kinds of mentions. Mentions in red are nominal, green are proper names, blue are pronominal. . . . 2
1.2 An example of context-dependent FGET. The left part is portion of a type taxonomy tree, the right part is a sentence with one mention being labelled with fine-grained type. . . . 2
1.3 An example of entity linking. . . . 3
1.4 An example of entity coreference resolution. . . . 3
1.5 Factor graph used as a joint model for NER, coreference resolution and entity linking [3]. . . . 4

2.1 Topic-Type relatedness and topic anchor. The colored parallelogram denote different topics. The words on the curves are three topic anchors that connect the entity type and topics. */Organization/company* is a type path on the type taxonomy. The colored squares are topic distributions, each square denotes one topic. . . . 22
2.2 Architecture of our FGET typing model based on transfer learning. The red rounded rectangle denotes the three transfer learning schemes. The dashed arrows denote the pre-training for transfer learning. . . . 24

3.1 Entity linking with embedded fine-grained semantic types. . . . 46
3.2 Local model for entity linking. . . . 48
3.3 Global model for entity linking [13]. . . . 50
3.4 Learning curves of **mulrel** [22] using two different sets of entity embeddings. . . . 59
3.5 T-SNE visualization of two sets of entity embeddings. Suffix "_wiki" denotes the Wikitext entity embeddings, while suffix "_sri" denotes the semantic reinforced entity embeddings ($T = 11, \alpha = 0.2$). . . . 59

4.1 The process of incorporating the coherence score between entity candidates and Anonymous Entity Type (AET) words (anonymous entity mentions). The AET words are highlighted. . . . 73
4.2 Building AET words inter-paragraph co-occurrence matrix. AET words are highlighted. . . . 75

5.1 The degree of anisotropic is measured by random similarity, the average cosine similarity between uniformly randomly sampled words. The higher the layer, the more anisotropic. Embeddings of layer 0 are the input layer word embeddings. (Figure 5.1-5.3 are generated using the method of Ethayarajh [12]) . . . 100
5.2 Intra-sentence similarities of contextualized representations. The intra-sentence similarity is the average cosine similarity between each word representation in a sentence and their mean. . . . 100

5.3 Self-similarities of contextualized representations. Self-similarity is the average cosine similarity between representations of the same word in different contexts. 104

List of Tables

2.1 Statistics of FGET corpus 30
2.2 Statistics of Topic Anchors 31
2.3 Hyperparameter settings 32
2.4 Performance on the FIGER(GOLD) and OntoNotes corpora 34
2.5 Statistics of testing examples in FGET corpora 35

3.1 F1 scores on six test sets. The last column is the average of F1 scores on the five out-domain test sets. 57

4.1 F1 scores on six test sets. The AIDA-B dataset is the in-domain test set, while the other five datasets are out-domain test sets. The methods in bold are our direct baselines. 80
4.2 Ablation analysis on the effectiveness of our proposed AET coherence scores. The "Average F1" denotes the averaged F1 on the five out-domain test sets. . . 81
4.3 Cosine similarity between "*investor*" and other AET words using different embeddings 82

5.1 Results on the test set of the OntoNotes English data from the CoNLL-2012 shared task. The rightmost column is the main evaluation metric, the average F1 of MUC, B^3, $CEAF_{\phi_4}$ 102
5.2 Performance on the test set of GAP corpus. The metrics are F1 scores on **M**asculine and **F**eminine examples, **O**verall F1 score, and a **B**ias factor(F/M). 103
5.3 Ablation studies of ELECTRA-base and SpanBERT-base. The metric is the average F1 score on the OntoNotes dev set and test set using different combinations of hyperparameters. The underlined numbers denote that the performance on the test set is even better than that on the dev set. The bold numbers denote the reported results in Table 5.1. 103

Abbreviations

AET	**A**nonymous **E**ntity **T**ype
BERT	**B**idirectional **E**ncoder **R**epresentations from **T**ransformers
CNN	**C**onvolutional **N**eural **N**etwork
CRF	**C**onditional **R**andom **F**ield
(E)CR	(**E**ntity) **C**oreference **R**esolution
(N)EL	(**N**amed) **E**ntity **L**inking
ELECTRA	**E**fficiently **L**earning an **E**ncoder that **C**lassifies **T**oken **R**eplacements **A**ccurately
FGET	**F**ine - **G**rained **E**ntity **T**yping
GRU	**G**ated **R**ecurrent **U**nit
HMM	**H**idden **M**arkov **M**odel
KB	**K**nowledge **B**ase
LASE	**L**ess - **A**nisotropic **S**pan **E**mbeddings
LBP	**L**oopy **B**elief **P**ropagation
LDA	**L**atent **D**irichlet **A**llocation
LSTM	**L**ong **S**hort - **T**erm **M**emory
MLP	**M**ulti - **L**ayer **P**erceptron
NED	**N**amed **E**ntity **D**isambiguation
NER	**N**amed **E**ntity **R**ecognition
NLP	**N**atural **L**anguage **P**rocessing
OOV	**O**ut **O**f **V**ocabulary
RNN	**R**ecurrent **N**eural **N**etwork
SVM	**S**upport **V**ector **M**achine

Chapter 1

Introduction

This chapter provides an overview of this thesis. We introduce the background of entity analysis in Section 1.1. We explain our motivation in Section 1.2, where the issues with the existing entity analysis approaches are analyzed. We present our research questions and hypotheses in Section 1.3, where the basic ideas for this thesis are proposed and explained. Thesis contributions are summarized in Section 1.4. The outline of this thesis is listed in Section 1.5.

1.1 Introduction to Entity Analysis

Identifying and understanding entity mentions are important for natural language understanding. Entity analysis is to detect and extract entity mentions and related information. However, entity analysis is challenging because of the ambiguity and coreference. The ambiguity exists in the complex relationship between the concrete entities and their natural language mentions. The same surface form of mention may refer to different entities, and the same entity may appears as different mentions. Moreover, a large proportion of information about entity is given by coreference (alias, nicknames, pronouns). Thus, it is reasonable to divide the entity analysis into smaller sub-tasks, with each sub-task tackling one aspect of entity analysis.

Several sub-tasks of entity analysis [3], [10] have been developed to characterize different aspects of entities and entity mentions. These sub-tasks are listed as follows.

Entity Mention Detection(EMD) [14],[19] is to identify all the mentions of entities, including proper names, noun phrases (nominals) and pronouns (pronominals) that refer to entities.

Appearing before Congress, Mr Mueller said he had not exonerated the president of obstruction of justice.

FIGURE 1.1: Entity mentions in a sentence are coloured. Different background colors denote different kinds of mentions. Mentions in red are nominal, green are proper names, blue are pronominal.

For example, EMD will detect the colored parts in Figure 1.1. Entity mention detection is the first step for other entity analysis tasks.

Named Entity Recognition (NER) [2] is to detect mentions of entities with proper names (i.e., named entities) and classify them into coarse-grained classes (typically 4 classes: *Person*, *Location*, *Organization* and *Miscellaneous* [20]). For example. NER will only identify the 'Mr Mueller' as *Person* in Figure 1.1.

Relationship Extraction [9] is to extract relational facts between entities from text, e.g., learning that a person is the head of a particular organization, or that an organization is located in a particular region.

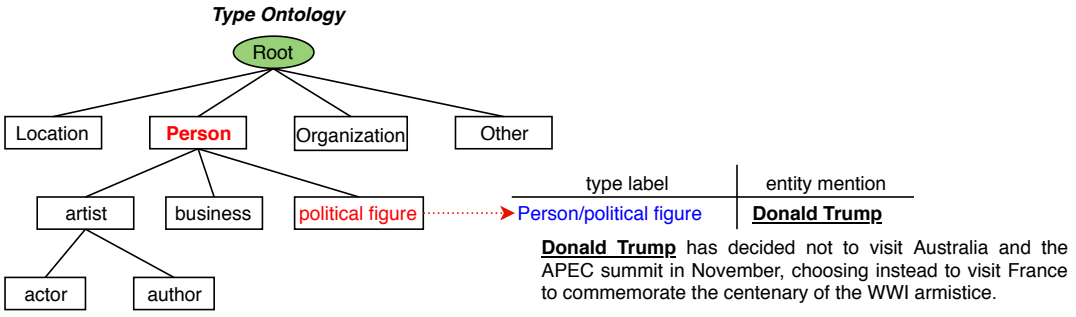

FIGURE 1.2: An example of context-dependent FGET. The left part is portion of a type taxonomy tree, the right part is a sentence with one mention being labelled with fine-grained type.

Fine-grained Entity Typing (FGET) [15] is to classify entity (mentions) into more fine-grained semantic types, e.g., *Person* is classified into more fine-grained types: *artist*, *political_figure*, etc. In Figure 1.2, the types are organized into a type taxonomy using a tree structure to represent the hyponymy and hypernymy relationship between types, and the mention *Donald Trump* is typed as */Person/political_figure*[1] according to the context. The fine-grained semantic information of entity mentions has been proven to be valuable for many entity analysis sub-tasks, such as entity linking [12], relation extraction [22], entity search [8] and coreference resolution [16].

[1]In this thesis, we use capitalized italic prints to denote level 1 type labels (e.g. */Person*), non-capitalized italic prints to denote level 2 and 3 types (e.g. */Person/artist*). The symbol "/" is used to represent the hierarchical relationship.

Entity Linking (EL) [18] or **Named Entity Disambiguation (NED)** is to link a named entity mention to the specific entity in a knowledge base it refer to in the context. For example, in Figure 1.3, the mention 'Mr Mueller' link to the special counsel for the U.S. Department of Justice and former FBI director according to the context.

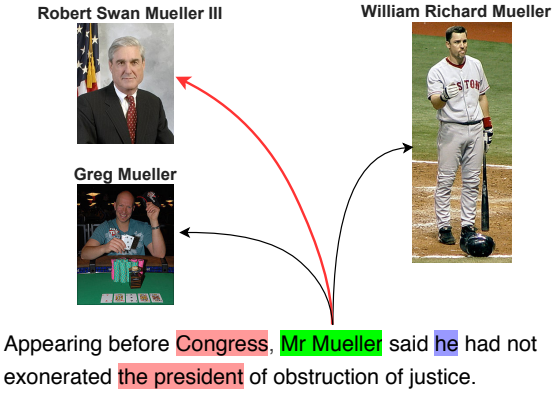

FIGURE 1.3: An example of entity linking.

Entity Coreference Resolution (ECR or CR) [21] is the task of deducing which entity mentions in neighbouring context refer to the same real world entity, and those coreferent mentions (names, noun phrase and pronoun) will be clustered. For example, in Figure 1.4, the mentions are clustered into two clusters, with each cluster being a collection of mentions that refer to the same entity.

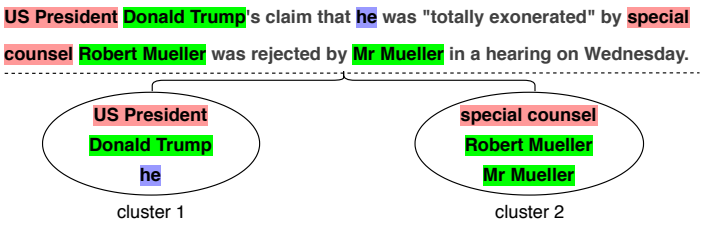

FIGURE 1.4: An example of entity coreference resolution.

These sub-tasks of entity analysis is highly interdependent, and some joint models for two or three of these sub-tasks have been proposed. For example, the joint model for NER and entity linking [11], the joint model for NER and relation extraction [7].

1.2 Motivation

Because the sub-tasks of entity analysis are highly interdependent. Joint models for entity analysis have been proposed to tackle two or more sub-tasks jointly to capture more information for making globally optimized decisions [11], [5], [3], [17]. These joint models are mainly

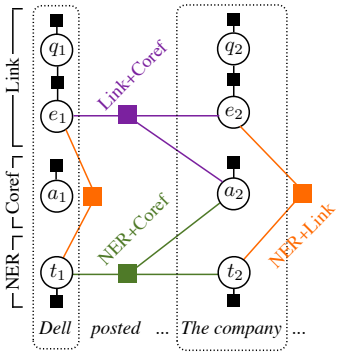

FIGURE 1.5: Factor graph used as a joint model for NER, coreference resolution and entity linking [3].

based on factor graph model [3], [17], and their experimental results show that joint models improve performance on all the subtasks incorporated.

Pantel et al. [13] proposed a graph based generative model to jointly model user intent and query entity types. The model is trained by maximizing the probability of observing a large collection of real-world queries and their clicked hosts. This method can only type the entities that appear in their web queries. Singh *et al.* [17] use factor graph model to represent the dependencies between entity typing, relation extraction and coreference resolution. Instead of training all the factors jointly, they use *piece-wise training* approach to estimate the parameters of the model. Parameters for each factor are learned independently by maximizing the piece-wise likelihood.

Durrett and Klein [3] tackle coreference resolution, entity typing and entity linking simultaneously using a conditional random fields (CRF) model. Unary factors define the features for solving each subtask independently. The binary and ternary factors define the features that capture the interactions or constraints between subtasks. The model is trained by maximizing the joint probability of three labels for all mentions in the corpus. However, for both learning and decoding, exact inference would be intractable because of the loops in the factor graph. Although belief propagation can perform efficient inference, it would still be computationally exorbitant due to the ternary factor. Thus they use a pre-trained coarse model to prune 90% of the possible coreference arcs. However, such a joint model require training corpus that have labels of all three subtasks, which is not readily available.

Although the previous factor graph model based joint models are able to capture the interactions between subtasks, they have the following issues:

- Cannot exploit document-level type correlatedness. Document-level type correlatedness refers to the coherence constraints of entity types in a document, e.g., *footballer* is more coherent with *football team* than with *defence contractor*. The collective model for FGET [15] takes into account correlations between entity mentions in a document, but only considers the heuristic coreference relations between entity mentions of a document. The document-level coherence constraints have been used in entity linking [4], but only the named entities are considered.

- Inefficient feature representations. The joint models rely on hand-crafted features. Such hand-crafted features are usually represented as one-hot high-dimensional vectors. Hence the models suffer from feature sparsity and the so-called 'curse of dimensionality'.

- Lack of jointly annotated corpora. The training of joint models need a corpus that has labels of three sub-tasks. However, such corpus is expensive to annotate. Currently only the ACE 2005 corpus[2] is available. With limited jointly annotated corpora, the performance of joint models is difficult to be further improved.

As we can see, joint models usually simultaneously tackle three sub-tasks: fine-grained entity linking, entity linking and coreference resolution. Recently, deep learning models for each of the three sub-tasks have achieved significant improvements by using learned feature embeddings. However, they are all independent models because of the lack of jointly annotated corpora, and the fine-grained type information of entities is not used for entity linking and coreference resolution.

We aim at improving FGET and entity linking by exploiting the document-level type constraints and improving entity linking and coreference resolution by embedding fine-grained entity type information. We present our research objectives and hypotheses in next section.

1.3 Research Objectives and Hypotheses

In this thesis, we will apply deep learning and transfer learning to three sub-tasks of entity analysis: fine-grained entity typing, entity linking and entity resolution. Our research objectives and hypotheses are listed as follows:

[2] https://catalog.ldc.upenn.edu/LDC2006T06

- **Transfer learning for fine-grained entity typing.** There are two main issues with existing FGET approaches. Firstly, the training corpora for FGET are normally labelled automatically, which inevitably induce noises. Existing approaches either directly tweak noisy labels in corpora by heuristics, or algorithmically retreat to parental types, both leading to coarse-grained type labels instead of fine-grained ones. Secondly, exist approaches usually use recurrent neural networks (RNN) to generate feature representations of mention phrases and their contexts, which, however, perform relatively poor on long contexts and out-of-vocabulary (OOV) words.

  We hypothesize that **there is a correlation between fine-grained types and hidden topics**. For example, noun *auto_maker/car_maker* is fine-grained types, and is related to *car_industry* topic. We learn this correlation using unlabelled documents. We explore transfer learning based approaches to extract more efficient feature representations and offset label noises. Especially, we use a pre-trained topic model to transfer the topic-type relatedness through topic anchors and select confusing fine-grained types at inference time. The pre-trained topic model can offset the label noises without retreating to coarse-grained types.

- **Entity linking with typed entity embeddings.** Neural entity linking models embed words and entities into a common dimensional space, and use entity embeddings as input to the local and global ranking score functions. If entity embeddings are too similar, it would be difficult for linking models to disambiguate similar entities. If entity embeddings are too distinctive, linking models cannot learn the contextual commonalities of similar entities. We argue that the current entity embeddings [4] learnt from Wikipedia articles encoded too many details of entities, thus are too distinctive for linking models to learn contextual commonality.

  We hypothesize that **fine-grained semantic types of entities can let the linking models learn contextual commonality about semantic relatedness**. For example, *rugby* related documents would have entities of *rugby player* and *rugby team*. If a linking model learns the contextual commonality of *rugby* related entities, it can correctly select entities of similar types using the similar contextual information. We explore methods of incorporating fine-grained type information into entity embeddings. Such methods include using embeddings of nouns of fine-grained semantic type, e.g. *'president', 'car_maker' 'actor'*.

- **Improving entity linking through anonymous entity mentions**. The existing entity linking methods of using global information exploit the information of candidate entities of named entity mentions (e.g., "Nardelli" and "Home Depot Inc"). However, such named entity mentions appear less frequently than anonymous entity mentions (e.g., *the company*). Thus, such methods can only use limited global information, but the more frequently occurring anonymous entity mentions are ignored. The anonymous entity mentions always appear as fine-grained entity type words (e.g., the *company, Canadian singer, service provider, news agency* etc.). These words are parts of anonymous entity mentions, and we call such words **Anonymous Entity Type** (**AET**) words.

  We hypothesize that **there exists the document-level entity type correlation**. For example, *company* and *chief executive* are highly related with each other in documents. Anonymous entity mentions usually appear as AET words, we can extract AET words in a document as anonymous entity mentions to infer the types of the named entity mentions. Thus, when ranking the candidate entities of "Nardelli", the entity "Robert Nardelli" with type *chief executive* is more coherent with the document that has many anonymous *company* mentions.

- **End-to-end entity coreference resolution based on fine-grained semantic types**. The end-to-end coreference resolution models tackle mention detection and coreference resolution simultaneously. They consider all spans as mention candidates. The core of end-to-end neural coreference resolution models is the learning of span embeddings. The state-of-the-art coreference resolution models use the output layer of a contextualization model to build span embeddings, and employ the document-level semantics to refine span embeddings as higher-order coreference resolution. However, the contextualized and higher-order refined span embeddings tend to be highly anisotropic (anisotropic embeddings are not directionally uniform, thus are more similar), and make it difficult to distinguish between related but distinct entities (e.g., *pilots* and *flight attendants*).

  We hypothesize that **less anisotropic span embeddings can improve the performance of end-to-end coreference resolution models**. This hypothesis is based on the finding that *less anisotropic* static word embeddings gain large improvements on downstream NLP tasks. We explore methods of injecting fine-grained type information into span embeddings to make them more distinctive. We also investigate LASE (**L**ess **A**nisotropic **S**pan **E**mbeddings) schemes to generate less anisotropic span embeddings.

1.4 Contributions

Throughout this thesis, we will focus on the following three sub-tasks of entity analysis: fine-grained entity typing (Chapter 2), entity linking (Chapter 3 and 4) and entity coreference resolution (Chapter 5). The contributions in each of the aforementioned chapters are summarized as follow:

1. Transfer learning for fine-grained entity typing.
   - We propose a novel transfer learning architecture that combines a non-recurrent neural language model and a topic model.
   - We show that topic model is capable of transferring the learned associations between semantic types and hidden topics.
   - We use sub-word patterns to generate the vectors of out-of-vocabulary (OOV) words that are constituents of entity mentions, and we use a pre-trained language model to encode context features.
2. Improving entity linking through typed entity embeddings.
   - We create a dictionary of fine-grained semantic type words.
   - We propose a method to inject fine-grained type information into entity embeddings.
   - We show that the typed (semantic reinforced) entity embeddings can let the linking models learn contextual commonality of similar entities.
3. Improving entity linking through anonymous entity type (AET) words/mentions.
   - We propose a novel method for extracting AET words' inter-paragraph co-occurrence and learning AET word embeddings where such embeddings can capture the relatedness of AET words from document-level context.
   - We incorporate a new coherence score based on AET entity embeddings and document's AET context embeddings.
   - We verify the effectiveness of the incorporated coherence score on standard benchmark datasets and achieve significant improvement over the baselines.
4. Exploiting less anisotropic span embeddings for coreference resolution.
   - We propose and investigate four LASE schemes to generate less anisotropic span embeddings for coreference resolution.

– Our extensive experiments show that: (i) When our **Internal** and **LowerDep** schemes are applied to ELECTRA [1] and SpanBERT [6], their performances are improved by +1.9 F1 and +0.5 F1 on the OntoNotes benchmark, respectively; (ii) The span embeddings from longer-context-encoded contextualized representations of ELECTRA and SpanBERT are more effective than higher-order span embeddings; (iii) The 12th layer embeddings of BERT-base are no better than the 11th layer embeddings for coreference resolution; (iv) The degree of anisotropy can be used as guidance for hyperparameter settings.

1.5 Thesis Outline

In Chapter 2, we present our research on transfer learning for FGET. We investigate three transfer learning schemes to extract more efficient feature representations and offset label noises.

In Chapter 3, we present our method of injecting fine-grained semantic type information into entity embeddings. We show that the semantic reinforced entity embeddings can let the linking models learn contextual commonality of similar entities.

In Chapter 4, we propose a method DOC-AET, to improve entity linking by exploiting the DOCument-level coherence of named entity mentions and anonymous entity type (AET) words/mentions.

In Chapter 5, we first analyze the sources of anisotropic span embeddings. We then propose LASE (**L**ess **A**nisotropic **S**pan **E**mbeddings) schemes and investigate their effectiveness for improving coreference resolution.

Chapter 6 contains our conclusion where we summarize our findings and discuss the future directions.

Note that references related to each chapter are listed at the end of each chapter.

References

[1] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *Proceedings of the*

8th International Conference on Learning Representations (ICLR), 2020. URL https://openreview.net/pdf?id=r1xMH1BtvB.

[2] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007.

[3] Greg Durrett and Dan Klein. A joint model for entity analysis: Coreference, typing, and linking. *Transactions of the association for computational linguistics*, 2:477–490, 2014.

[4] Octavian-Eugen Ganea and Thomas Hofmann. Deep joint entity disambiguation with local neural attention. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2619–2629, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1277. URL https://www.aclweb.org/anthology/D17-1277.

[5] Gang Luo, Xiaojiang Huang, Chin-Yew Lin, and Zaiqing Nie. Joint named entity recognition and disambiguation. In *Proceedings ofthe 2015 Conference on Empirical Methods in Natural Language Processing*, pages 879–888, 2015.

[6] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77, 2020.

[7] Qi Li and Heng Ji. Incremental joint extraction of entity mentions and relations. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 402–412, Baltimore, Maryland, June 2014. Association for Computational Linguistics. doi: 10.3115/v1/P14-1038. URL https://www.aclweb.org/anthology/P14-1038.

[8] Denghao Ma, Yueguo Chen, Kevin Chen-Chuan Chang, Xiaoyong Du, Chuanfei Xu, and Yi Chang. Leveraging fine-grained wikipedia categories for entity search. In *Proceedings of the 2018 World Wide Web Conference*, pages 1623–1632, 2018.

[9] Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011, 2009.

[10] David Nadeau. *Semi-supervised named entity recognition: learning to recognize 100 entity types with little supervision*. PhD thesis, University of Ottawa, 2007.

[11] Dat Ba Nguyen, Martin Theobald, and Gerhard Weikum. J-NERD: joint named entity recognition and disambiguation with rich linguistic features. volume 4, pages 215–229. MIT Press, 2016.

[12] Nitish Gupta, Sameer Singh, and Dan Roth. Entity linking via joint encoding of types, descriptions, and context. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 2671–2680, 2017-09-07.

[13] Patrick Pantel, Thomas Lin, and Michael Gamon. Mining entity types from query logs via user intent modeling. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 563–571. Association for Computational Linguistics, 2012.

[14] Radu Florian, Hongyan Jing, Nanda Kambhatla, and Imed Zitouni. Factorizing complex models: a case study in mention detection. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, 2006-07-17.

[15] Altaf Rahman and Vincent Ng. Inducing fine-grained semantic classes via hierarchical and collective classification. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 931–939, Beijing, China, August 2010. Coling 2010 Organizing Committee. URL https://www.aclweb.org/anthology/C10-1105.

[16] Marta Recasens, Marie-Catherine de Marneffe, and Christopher Potts. The life and death of discourse entities: Identifying singleton mentions. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 627–633, Atlanta, Georgia, June 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/N13-1071.

[17] Sameer Singh, Sebastian Riedel, Brian Martin, Jiaping Zheng, and Andrew McCallum. Joint inference of entities, relations, and coreference. In *Proceedings of the 2013 Workshop on Automated Knowledge Base Construction*, 2013.

[18] Wei Shen, Jianyong Wang, and Jiawei Han. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering*, 27 (2):443–460, 2014.

[19] Thien Huu Nguyen, Avirup Sil, Georgiana Dinu, and Radu Florian. Toward mention detection robustness with recurrent neural networks. In *Proceedings of IJCAI Workshop on Deep Learning for Artificial Intelligence*, 2016.

[20] Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147, 2003. URL https://www.aclweb.org/anthology/W03-0419.

[21] Vincent Ng. Machine learning for enity coreference resolution: A retrospective look at two

decades of research. In *Proceedings of the 31st AAAI conference on Artificial Intelligence*, pages 4877–4884, 2017.

[22] Yadollah Yaghoobzadeh, Heike Adel, and Hinrich Schütze. Noise mitigation for neural entity typing and relation extraction. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1183–1194, Valencia, Spain, April 2017. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/E17-1111.

Chapter 2

Transfer Learning for Fine-grained Entity Typing

Fine-Grained Entity Typing (FGET) is to classify the mentions of entities into hierarchical fine-grained semantic types. There are two main issues with existing FGET approaches. Firstly, the process of training corpora for FGET is normally to label the data automatically, which inevitably induces noises. Existing approaches either directly tweak noisy labels in corpora by heuristics, or algorithmically retreat to parental types, both leading to coarse-grained type labels instead of fine-grained ones. Secondly, exist approaches usually use recurrent neural networks (RNN) to generate feature representations of mention phrases and their contexts, which, however, perform relatively poor on long contexts and out-of-vocabulary (OOV) words. In this chapter, we propose a transfer learning based approach to extract more efficient feature representations and offset label noises. More precisely, we adopt three transfer learning schemes: (i) transferring subword embeddings to generate more efficient OOV embeddings; (ii) using a pre-trained language model to generate more efficient context features; (iii) using a pre-trained topic model to transfer the topic-type relatedness through topic anchors and select confusing fine-grained types at inference time. The pre-trained topic model can offset the label noises without retreating to coarse-grained types. The experimental results demonstrate the effectiveness of our transfer learning approach for FGET.

2.1 Introduction

Identifying and classifying the mentions of entities in text are important for Natural Language Understanding (NLU). The traditional coarse-grained Named Entity Recognition (NER) detects the boundaries of entity mentions and classifies them into four coarse-grained types, i.e. *Person*, *Location*, *Organization* and *Miscellaneous*. However, in order to enhance the performance, most Natural Language Processing (NLP) tasks need more fine-grained semantic information about those entity mentions, such as */Person/artist/actor* [1]. For example, the task of entity linking is to disambiguate entity mentions and link the mentions to a specific entity in a knowledge base, e.g., link a name to a particular person. Most knowledge bases have fine-grained semantic tags (i.e. *actor, author*) for the majority of the entities they stored. If a similar fine-grained semantic type is gained from the context around the entity mentions, entity linking will drastically reduce the number of candidate entities by selecting only the entities that have the same fine-grained semantic type. The fine-grained semantic information of entity mentions has been proven to be valuable for many NLP tasks, such as entity linking [36], relation extraction [58], entity search [30] and coreference resolution [44].

Different fine-grained type taxonomies have been proposed for different scenarios. The number of types varies from less than one hundred to over one thousand. The most widely used type taxonomies are FIGER [56] and GFT [12], which have 112 and 88 types respectively. The typing model that classifies entity mentions into the type labels on the type taxonomies is generally formulated as a hierarchical classifier based on machine learning methods.

Normally, the training data for FGET are automatically labelled with entity linking tools, such as Dbpedia Spotlight [32]. The semantic tags in the knowledge base are then mapped to the type taxonomy. This inevitably induces label noises in the training data. There are mainly two approaches to address this issue. One approach is to use heuristics to preprocess the training data. Such heuristics usually remove less frequent types in a document or more fine-grained types. As a result, the preprocessing makes the training corpus skewed toward coarse-grained type labels [12]. The other approach is to conservatively discourage the typing model from predicting more fine-grained types [37]. The typing model usually makes predictions on the features extracted from the context and the words that consist the entity mention. The features used by traditional machine learning algorithms are binary feature functions

[1] In this thesis, we use capitalized italic prints to denote level 1 type labels (e.g. */Person*), non-capitalized italic prints to denote level 2 and 3 types (e.g. */Person/artist*). The symbol "/" is used to represent the hierarchical relationship.

[42], [34], while the features used by deep learning methods are embedded dense dimensional vectors generated by recurrent neural networks [49], [37], such as Long Short-Term Memory (LSTM) [22]. The former suffers from feature sparsity, while the latter suffers from insufficient feature representations. These features are extracted from the local context of a small window, while the document-level features are seldom used.

To address those aforementioned issues in FGET, we propose a novel transfer learning based approach, the main contribution of which can be summarized as follows:

- We introduce a novel transfer learning architecture that combines a non-recurrent neural language model and a topic model. Our transfer learning architecture is different from the recent transfer learning methods being used in NLP tasks that mainly focus on language models [26], [60], [23], [40], [38]. The intuition behind is that the language model is able to capture the local context, while the topic model is able to find the distributions of words and semantic classes across documents and latent topics.
- We show that the topic model is capable of transferring the learned associations between semantic types and hidden topics. The document level topic label has been used as a feature for FGET [12], [61], [31] or used to reduce label noises [12]. But their topics are obtained through a supervised text classification model. However, in the new approach, we use the learned topic model to guide the inference of a typing model instead. Such mechanism can reduce the confusion caused by label noises without retreating to coarse-grained types.
- We use transfer learning to generate the embedding vectors of out-of-vocabulary (OOV) words that are constituents of entity mentions. Previous work on FGET learned the embedding vectors of OOVs during the training of the FGET classifier. This leads to insufficient training, and the sub-word level information of OOVs are ignored. We use the sub-word level patterns to estimate the embeddings of OOVs based on embeddings of character n-grams.
- We use a pre-trained language model to encode context features. Most of the previous work used LSTM trained on FGET data set with label noises. This makes the LSTM network incompetent at encoding effective context features. The deep neural language model pre-trained on clean corpus can generate more efficient feature representations.

The rest of this chapter is organized as follows. We review related works in Section 2.2. In Section 2.3, we give definitions about the mention-level FGET. In Section 2.4, we explain the topic-type relatedness hypothesis, which is the basis of using a topic model as transfer learning scheme for FGET. In Section 2.5, we present the new approach in detail. In Section 2.6, we report and analyze our experimental results on the datasets that are commonly used for FGET evaluation. We finally conclude our work in Section 2.7.

2.2 Related Work

Our research primarily relates to three domains: the fine-grained entity typing, transfer learning and the vector representation of words. In this section, we review related literature on these three domains.

2.2.1 Fine-Grained Entity Typing

There are two categories of FGET: The *mention-level* FGET [12] and the *entity-level* FGET [57]. The former is to determine the semantic type of an entity mention in a particular context, while the latter is to find all the possible semantic types of an entity. For example, *Donald J. Trump* can be a */Person/political_figure* or */Person/business* in different contexts, thus the set of entity-level types of *Donald J. Trump* includes the aforementioned two types. Our research falls under the mention-level FGET.

FGET is typically a multi-class multi-label classification problem. It is also a hierarchical classification problem because of the hierarchical relations among the type labels. According to the categorization of hierarchical classifiers by [9], there are mainly four categories of classifiers used in FGET:

- **Flat**: Using a single multi-class classifier for all types. Such classifiers mainly include: decision tree [33], linear classifier [56], [35], [61], [62], [2], and softmax classifier [12], [28], [20]
- **Local**: Using a binary classifier for each type, enforcing label consistency at inference time. Such classifiers include: SVM [34], maximum entropy classifier or logistic regression classifier [16], [12], [31].

- **Local per Parent Node**: Using multi-class classifier for all children of a parent type node. The RNN encoder-decoder based typing model of [48] falls into this category.
- **Global**: Using a single multi-class classifier trained with loss function that considers label similarity and hierarchical relationships. The typing models of [37], [39] fall into this category.

The feature representations have significant influence on the performance of classifiers. The features used by traditional machine learning methods [56], [12] are hand-crafted binary feature functions. To combat the feature sparsity, some methods [61], [62] embed these binary features into low-dimensional vectors. The neural network models for FGET usually employ different neural networks (with different parameters) to embed the mention features and context features. Most of the methods [37], [49], [2] use LSTM to encode the mentions and contexts.

2.2.2 Transfer Learning in NLP

There are two categories of transfer learning in NLP tasks: *resource-based* and *model-based* transfer learning [59], [5]. Resource-based transfer learning resorts to additional annotated resources (e.g. cross-lingual dictionaries) as weak supervision. Model-based transfer learning exploits the relatedness and similarity between the source task and target task. Domain adaptation [13] and multi-task learning [11] are the popular model-based transfer learning paradigms. Our work on transfer learning for FGET mainly focuses on transfer through a pre-trained language model and a topic model.

2.2.2.1 Transfer Learning Through Pre-trained Language Models

Very recently, the state-of-the-art results of a lot of NLP tasks have been achieved through transfer learning from neural language models. The neural network models for target tasks are partially pre-trained on a language model objective before fine-tuning on the supervised data set. Such contextualized representations from pre-trained language model include: ELMo[38], GPT [40], BERT [23], XLNet [60], and the very recently released SpanBERT [26] and ELECTRA [10] . The model architectures of GPT, BERT,XLNet, SpanBERT and ELECTRA are based on Transfomers [53], which has been used ubiquitously in recent research. Transformer

is a neural network architecture that is based solely on attention mechanisms and dispensed with recurrence and convolutions.

2.2.2.2 Transfer Learning Through Topic Models

Topic model has been used to Multi-label dataless text classification [63]. Wiedemann et al. [54] used the tweets annotated with topic clusters of LDA (Latent Dirichlet Allocation) [6] as an additional training dataset to improve the performance of offensive language detection. Deng et al. [14] used a topic model to capture the sentiment polarity of word in different topics and construct a domain-specific sentiment lexicon, which was used to improve the performance of sentiment classification. Baheti et al. [4] introduced semantic and topic similarity constraints in the decoding objective to generate more content rich responses for a dialogue system. The topic similarity between source and response is based on the topic distribution of the sources and responses. Jin et al. [25] proposed a model named *LSTM-Topic matrix factorization* that combines LSTM and a topic model for review understanding.

2.2.3 Vector Representation of Words

Word vectors have been successfully used in neural network models for NLP tasks. The *prediction-based* word vectors, such as Word2Vec [51], [52], are learned through training a language model to predict context words. The *count-based* word vectors, such as GloVe [24], are learned based on the word co-occurrence matrix. These two embedding paradigms cannot generate the vectors of OOVs, the words that do not appear in the training corpus. The *fast*Text [7] method introduced the sub-word information to the Skip-Gram model of the Word2Vec [52], thus is capable of generating word vectors for OOVs. Sub-word representations are essentially useful for modeling rare words and OOVs. The subword-augmented embeddings significantly improved performance on text understanding tasks [65].

2.3 Mention-level FGET Task Definition

Our work focuses on mention-level FGET with transfer learning. In this section, we present how we formulate the mention-level FGET by giving several definitions.

Entity Mention: Entity mention is a continuous span of tokens in the text which refers to a real world entity. Entity mention can be a named entity mention, a nominal mention or

pronoun reference. For example, the *Jaguar* in the following sentence: *The engine plant may encompass plans for a joint components venture with Jaguar.*

Type Taxonomy: To naturally represent the hierarchical structure of the semantic types, type taxonomy or type ontology is defined as a tree or a directed acyclic graph (DAG) $O = (T, R)$, where T is the set of semantic types and R is the edge set. $R = \{(t_i, t_j) \mid t_i, t_j \in T,\ i \neq j\}$ is also called the relation set, in which (t_i, t_j) means that t_j is a finer-grained sub-type of t_i.

Mention-level FGET: Mention-level FGET can be defined as $f : M \times C \mapsto T$(where C is the set of corresponding context of each mention in M), which is to find a semantic type with appropriate degree of granularity for an entity mention that appears in a specific context. The appropriate degree of granularity means that the semantic types should be inferred from the context, and should not be too specific or too general. Mention-level FGET is also called context-dependent FGET.

We formulate the typing model for FGET as a local binary classifier on each type of the taxonomy. The typing model makes predictions based on the feature representations of mention phrase and context. For each mention m, we denote the embedded mention feature as $\mathbf{v}_m$, and the context feature as $\mathbf{v}_c$. The probability of entity mention m being type t_i can be computed by:

$$P(t_i|m, c) = \sigma(f_i(\mathbf{v}_m \oplus \mathbf{v}_c) + b) \tag{2.1}$$

where σ is the *sigmoid* function, b is bias.

Previous methods for embedding the $\mathbf{v}_m$ and $\mathbf{v}_c$ are based on LSTM, which is unable to capture effective context in long sentences. We use model-based transfer learning to get more efficient feature representation of v_m and v_c. We will describe this in more details in Section 2.5.

Label Noises: The automatically generated training examples have two kinds of label noises: the out-of-context noise and the overly-specific noise [37]. To explain this, consider the mention *Hugh Laurie* in the following two sentences: (i) *Hugh Laurie and his wife Jo Green were on the verge of divorce.* (ii) *Hugh Laurie wins the Best Supporting Actor in a miniseries. Hugh Laurie* has multiple labels (e.g. *actor, director, musician, comedian*, and *author*) in a knowledge base, and all these labels will be assigned to both mentions in both sentences. In sentence (i), the type label should be *Person*, while all other types are overly-specific label noises. In sentence

(ii), the type label should be *Person/artist/actor*, while all the other types are out-of-context label noises.

To tackle the label noises, previous research either directly tweaks the noisy labels using heuristics, such as [12], [17], [1], or enhances the typing model with the ability of tackling label noises, such as [45], [46], [2], [37], [29]. The former approach makes the training corpus skewed toward coarse-grained labels, while the latter approach encourages the typing model to predict relevant parent-types. Thus both approaches tend to retreat to coarse-grained types. We train our typing classifier on intact noisy corpus. Then we select multiple labels based on the probability and filter out those irrelevant labels using topic-type relatedness. We provide the details of topic-type relatedness in Section 2.4.

2.4 Topic-Type Relatedness Hypothesis

The training data sets for FGET are peppered with quite a few label noises. Thus the trained typing model is still confused on many predictions, no matter how effective the features learned through transferred model are. To alleviate the confusion caused by the label noises, we propose to post-process the outputs of the typing model by exploiting the topic-type relatedness at inference time. In this subsection, we introduce our hypothesis about topic-type relatedness.

2.4.1 Topic-Type Relatedness

Unlike the text classification, which is a supervised NLP task of classifying text documents into relatively small number (less than 15) of topics (relatively coarse-grained topics, such as *politics, sports, business* etc.), topic model [6] is an unsupervised generative model that treats each document as a random mixture of latent topics, where a topic is defined as a multinomial distribution over words in vocabulary. The number of topic is set as a hyperparameter. A large number of topics can enable the topic model to capture more fine-grained topics. The unsupervision character of the topic model means it can be trained on a relatively large and clean corpus.

Our hypothesis is that there are connections or dependencies between latent topics and semantic types. Semantic types of entities are usually directly provided with an appositional structure. Sometimes only abstract semantic type is mentioned without the concrete entity.

Although sometimes the appositional semantic types are absent, we can still capture the relatedness between the semantic types and latent topics through the appositions [2].

Our hypothesis is based on the hypotheses of the LDA (Latent Dirichlet Allocation) topic model [6] and the class-based n-gram model [8]. The LDA hypothesizes that documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words. The class-based n-gram model assume that the word class of prediction words is conditioned on the word classes of $n-1$ histories, i.e., the probability of a string of words, w_1^k, is computed by $Pr(w_k|w_1^{k-1}) = Pr(w_k|c_k)Pr(c_k|c_1^{k-1})$, where c_i is the word class of word w_i. Thus, the generative process for each document can be assumed as follows:

1. Choose the number of words $N \sim \text{Poisson}(\xi)$.
2. Choose a topic distribution $\theta \sim \text{Dir}(\alpha)$.
3. For each of the N words w_n:
   (i) Choose a topic $z_n \sim \text{Multinomial}(\theta)$.
   (ii) Choose a semantic class s_n from $p(s_n|z_n, \beta)$.
   (iii) Choose a word (or concrete entity mention) from semantic class s_n.

For example, in a sentence of *car industry* topic, when *automaker* type is chosen, the word could be chose from {*car_maker*, *automaker*, *Jaguar*, *Toyota*, ...}. We define the topic-type relatedness as follows.

Topic-Type Relatedness: *Topic-type relatedness is the statistical dependence/association between a latent topic and the semantic types of entity mentions appear in the documents of that topic.*

For example, a *car_maker* entity is more likely to appear in a sentence that is about *car industry* topic; a *law_firm* is more likely to be associated with a *legal*-related topic. Inserting entities of other semantic types may causes incoherent with context or drastic change of the topic distribution.

To illustrate the topic-type relatedness more concretely, consider the following three sentences retrieved from the OntoNotes [43] corpus:

[2] Two noun phrases are placed next to each other to identify the same entity in a different way, e.g., Hugh Laurie, an award winning actor.

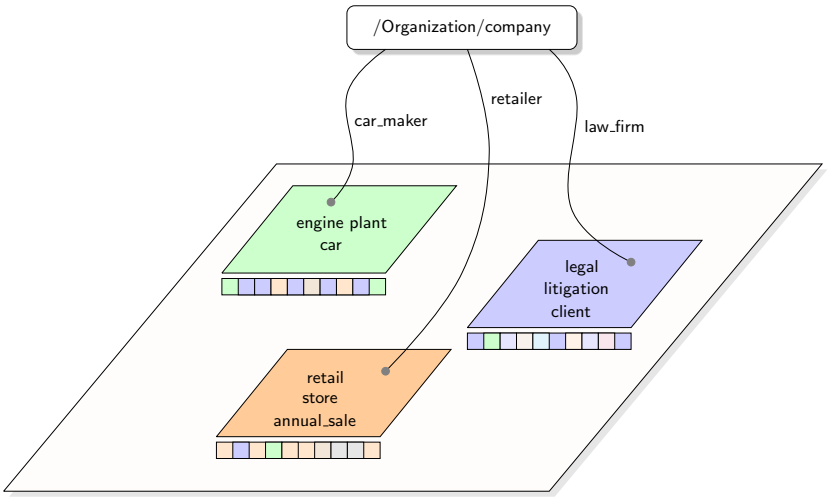

FIGURE 2.1: Topic-Type relatedness and topic anchor. The colored parallelogram denote different topics. The words on the curves are three topic anchors that connect the entity type and topics. */Organization/company* is a type path on the type taxonomy. The colored squares are topic distributions, each square denotes one topic.

A: *But the court jester appears to be Japan's smallest car maker, Daihatsu Motor Co.*

B: *Alternatively, a separate engine plant may be built as part of GM's planned tie-up with the British luxury car maker, the sources said.*

C: *The engine plant may encompass plans for a joint components venture with Jaguar.*

Sentence A directly provided the semantic class of entity mention *Daihatsu Motor Co.* using an appositional structure. Sentence B anonymously mentioned an entity whose semantic class is *car_maker*. Sentence C mentioned an entity named *Jaguar* without semantic class information. These three sentences have similar topic — *car industry*. We can apprehend the connection between the *car industry* topic and the semantic type *car_maker* using the first two sentences, then we can predict the entity mention *Jaguar* in Sentence C to be *car_maker* based on such analogy.

2.4.2 Topic Anchor

However, we cannot associate the topics directly with the semantic types on the type taxonomy. Because the types on the taxonomy, e.g. */Organization/company*, usually do not appear as apposition with the entity mentions. Instead, more fine-grained types (e.g. *car_maker*) are used as apposition. Thus, we use topic anchors to associate the types and topics.

Topic Anchor: *A semantically more fine-grained hyponym of a particular type that appears frequently in corpus and associates the hypernym type on the taxonomy and a topic.*

In Figure 2.1, three *topic anchors* that associate the type */Organization/company* with three corresponding topics are labelled on the curves. The words in the three coloured parallelograms are the keywords of corresponding topics.

The reason for combining a pre-trained language model and a topic model to transfer learning is that we believe a topic model can capture the relatedness between semantic classes and topics through some key words (i.e. *engine plant*), while the pre-trained language model can encode the context. We use the topic similarity between two topic distributions to rank the candidate topic anchors. One topic distribution is estimated solely on contextual words without topic anchor. The other topic distribution is estimated on the combination of contextual words and topic anchors. We will describe in more details in Section 2.5.

2.5 Approach

In this section, we introduce a new transfer learning based approach for FGET. We first present the framework of our typing model for FGET, and then we give details about the following three transfer learning schemes: (i) Using sub-word embeddings to to generate the embeddings of out-of-vocabulary (OOV) words in entity mentions; (ii) Using a pre-trained language model to encode context features; (iii) Using a pre-trained topic model to guide the inference of the typing classifier.

2.5.1 Typing Model

The architecture of our transfer learning based FGET typing model is shown in Fig. 2.2. According to the previous research [56], [34], [12], [49], the performance of local binary classifiers on FGET is more promising than the other classifiers introduced in Section 2.2.1, thus we use the local binary classifier. A local binary classifier makes predictions based on the feature representations of mention phrases and context words. The feature representations will be generated through model-based transfer learning as described in Section 2.5.2.

We implement our local binary classifiers using multi-layer perceptron (MLP) with one hidden layer. All the local binary classifiers share the parameters between the input layer and hidden

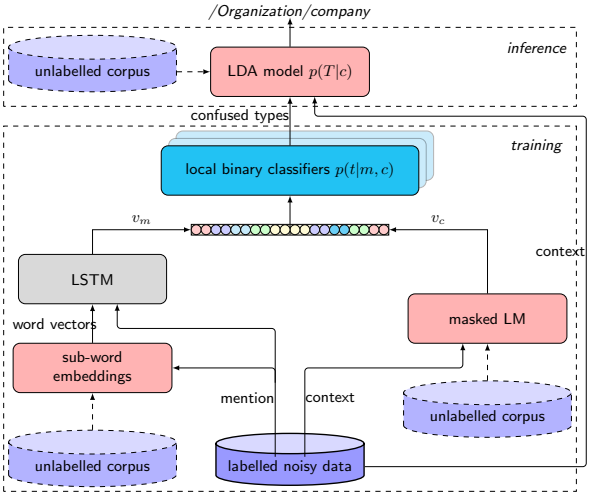

FIGURE 2.2: Architecture of our FGET typing model based on transfer learning. The red rounded rectangle denotes the three transfer learning schemes. The dashed arrows denote the pre-training for transfer learning.

layer. The parameters between the hidden layer and the output layer are parameters for the local binary classifiers. We also apply dropout training [21], [50] to our typing model. The training for such local classifiers needs to select negative and positive examples for each local classifier. For type label A, we simply treat all the other examples that do not have label A as negative examples. We use the sigmoid cross entropy with logits[3] [18], [27] as the loss function for training. We use $\hat{\mathbf{t}}$ to denote the vector of probabilities of mention m in context c being each type t, $\mathbf{t}$ denote the one-hot vector of true labels. $\hat{\mathbf{t}}$ is computed using Equation (2.2). We use Equation (2.3) to compute the loss function.

$$\hat{\mathbf{t}} = \sigma(((\mathbf{v}_m \oplus \mathbf{v}_c)\mathbf{W}_h + \mathbf{b}_h)\mathbf{W}_o + \mathbf{b}_o) \tag{2.2}$$

$$\mathbf{L}_{scewl} = -\mathbf{t} \log \hat{\mathbf{t}} - (1 - \mathbf{t}) \log (1 - \hat{\mathbf{t}}) \tag{2.3}$$

where $\mathbf{W}_h$ and $\mathbf{W}_o$ are the parameters of hidden layer and output layer, respectively; $\mathbf{b}_h$ and $\mathbf{b}_o$ are the bias for hidden layer and output layer, respectively; $\mathbf{L}_{scewl}$ denotes the *s*igmoid *c*ross *e*ntropy *w*ith *l*ogits loss function.

At inference time, we run the local classifiers simultaneously and independently select those type labels whose estimated probability is above a threshold. To improve the recall of types, we use a relatively small threshold. We then employ the following inference strategy to assign type labels:

[3] https://www.tensorflow.org/api_docs/python/tf/nn/sigmoid_cross_entropy_with_logits

(i) If there are neither level 2 nor level 3 fine-grained types, then select the level 1 coarse-grained type with highest probability.

(ii) If there are only one level 2 type, then select the types on the label path.

(iii) If there are multiple level 2 types, then use pre-trained topic model to filter out irrelevant types. The algorithm is described in more details in Section 2.5.3.

2.5.2 Transfer Learning for Feature Representations

In this subsection, we propose two transfer learning schemes for encoding mention features and context features, respectively.

2.5.2.1 Transfer Learning for Mention Embedding

The character-level patterns of words in mention phrases may provide strong typing information. For example, the word with "-shire" suffix, such as *Berkshire*, almost certainly is a *Location*. There are many OOVs in the entity mentions in training and testing corpus. The previous methods usually apply learning on-site to such OOVs. To generate more precise vector representations of these OOVs, we use transfer learning to generate OOV word vectors based on the sub-word information. Specifically, we use the *fast*Text [7] word embeddings to produce embeddings of OOVs. For example, we let the OOV *Berkshire* has a similar vector to *Hampshire*, whose word vector are trained on large corpus.

To capture the internal structure of an entity mention that has more than one word, we use single-directional LSTM to encode such information. We combine the averaging encoder and LSTM encoder to generate the mention embedding. Given mention m with $|m|$ words, the last output $h_{|m|}$ of LSTM encoder act as the LSTM representation of m, i.e. $\mathbf{v}_m^l = h_{|m|}$. The averaging embedding is computed as follow.

$$\mathbf{v}_m^a = \frac{1}{|m|} \sum_{i=1}^{|m|} v_i \tag{2.4}$$

where v_i is the vector of ith word of mention m obtained through *fast*Text. The embedded representation $\mathbf{v}_m$ of mention m is the concatenation of LSTM embedding $\mathbf{v}_m^l$ and averaging embedding $\mathbf{v}_m^a$.

$$\mathbf{v}_m = \mathbf{v}_m^l \oplus \mathbf{v}_m^a \tag{2.5}$$

2.5.2.2 Transfer Learning for Context Embedding

Context representation captures the information about the context surrounding the mention phrase. We use a pre-trained language model to generate the embedded context representation. Specifically, we use the SpanBERT (Span **B**idirectional **E**ncoder **R**epresentations from **T**ransformers) model [26], [23], which was trained on huge corpus through the so-called MLM (Masked Language Modeling) and SBO (Span Boundary Objective) objective (i.e. predicting the randomly masked span based on the left and right contexts). Unlike the left-to-right language model pre-training, the span prediction based on bidirectional context representations allows for pre-training of a deep bidirectional Transformer [53] to better represent and predict spans of text. Before transferring the SpanBERT model, we also fine-tune it on the CoNLL 2003 NER dataset [47] whose named entities have been annotated with coarse-grained types, i.e. *Person, Location, Organization and Miscellaneous*. For each sentence, the SpanBERT adds a special token ([CLS]) at the head and tokenize the sentence with WordPiece [55] vocabulary.

The SpanBERT model is based on a bi-directional full attention mechanism, thus the encoding of each token captures contextual information through attentions. We use the top layer hidden states of mention words to generate contextual embedding. Suppose the original mention phrase with $|m|$ words align with l tokens in the tokenized sentence, and the top layer hidden representation of each token is T_i. Then the contextual feature representation is computed as follow:

$$\mathbf{v}_c = \frac{1}{l}\sum_{i=1}^{l} T_i \tag{2.6}$$

2.5.3 Transfer Learning Through Topic Model

In this subsection, we give details about using a pre-trained topic model at inference time to select types. We use the HMM-LDA model to capture the topic-type relatedness, and use the similarity of topic distributions to measure the coherence between a type and context. For example, suppose the typing classifier outputs two type labels for mention Jaguar: *Organization/company* and *Organization/education*; *carmaker* and *university* are topic anchors of type label *Organization/company* and *Organization/education* respectively; if *carmaker* is

more coherent with the context than *university* (i.e., the topic distribution of *carmaker* is more similar with the topic distribution of context words), then the entity mention should be typed as *Organization/company*.

2.5.3.1 HMM-LDA Model

Griffiths et al. [19] proposed the HMM-LDA model to distinguish between function and content words. The HMM-LDA is a composite model, in which the syntactic component is an HMM and the semantic component is a topic model. Each component divides words into finer groups according to a different criterion: the function words are divided into syntactic classes, and the content words are divided into semantic topics. The HMM determines when to generate a semantic word from a LDA topic model and when to generate a function word from syntactic classes. The LDA chooses which content word to emit.

The HMM-LDA model is defined in terms of three sets of variables: a sequence of words $\mathbf{w} = \{w_1, ..., w_n\}$, with each w_i being one of W words, a sequence of topic assignments $\mathbf{z} = \{z_1, ..., z_n\}$, with each z_i being one of T topics, and a sequence of syntactic classes $\mathbf{c} = \{c_1, ..., c_n\}$, with each c_i being one of C classes. One class, $c_i = 1$, is assigned to content words that are drawn from topics. The zth topic is associated with a distribution over words $\phi^{(z)}$, each syntactic class $c \neq 1$ is associated with a distribution over words $\phi^{(c)}$, each document d has a distribution over topics $\theta^{(d)}$, and transitions between classes c_{i-1} and c_i follow a distribution $\pi^{(s_{i-1})}$. A document is generated via the following procedure:

1. Sample $\theta^{(d)}$ from a Dirichlet(α) prior

2. For each word w_i in document d

   (i) Draw z_i from $\theta^{(d)}$

   (ii) Draw c_i from $\pi^{(c_{i-1})}$

   (iii) If $c_i = 1$, then draw w_i from $\phi^{(z_i)}$, else draw w_i from $\phi^{(c_i)}$

2.5.3.2 Capturing Topic-Type Relatedness with HMM-LDA

To reduce the noise of non-semantic words, we employ the HMM-LDA [19] model to capture the topic-type relatedness.

The HMM-LDA model can approximately implement the generative process of our hypothesis in Section 2.4.1. The reasons can be explained as follows:

- Our hypothesis only aims at capturing the topic of semantic words. In the mean time, the HMM component can separate the function words from semantic (content) words.
- Our hypothesis only aims at capturing the topic-type relatedness, where the types are fine-grained semantic words, e.g., *carmaker*, *university*, etc. When we treat each of such semantic words as a semantic class of a topic (i.e., each semantic class has one word), the LDA component can learn the association between topics and semantic words (classes).

Markov Chain Monte Carlo inference is applied to estimate our model parameters on a corpus. We apply collapsed Gibbs sampling to draw iteratively a topic assignment and class assignment for each word. Each sample is used to estimate the model parameters after a burn-in of 2,000 iterations.

2.5.3.3 Topic Distribution Estimation

We use the similarity of topic distributions to measure the degree of semantic coherence of entity types with context. Given the HMM-LDA model's parameters, a simple but efficient method of estimating topic distributions is averaging topic distributions over all words in context. But, to ameliorate the non-semantic noises, we employ the probability of the word being emitted by the LDA topic model (a special syntactic class $c = 1$) $P(c = 1|w)$ as a weight to compute the topic distribution. If a word is topic anchor, then we fix the weight to be 1. The topic distribution is computed as follows:

$$P(T|C) = \frac{1}{Z} \sum_{w \in C} P(T|w) f_w \tag{2.7}$$

$$f_w = \begin{cases} 1, & \text{if } w \in A \\ P(c = 1|w), & \text{otherwise} \end{cases} \tag{2.8}$$

where $Z = \sum_{w \in C} f_w$ is a constant for normalization.

2.5.3.4 Type Selection

For level 2 or 3 type t (e.g. *Organization/company*), we use A_t to denote the collection of its topic anchors. For each topic anchor $a \in A_t$, we estimate the topic distribution based on the semantic words of context C plus the topic anchor. We then compare this topic distribution with the topic distribution estimated without the topic anchor. The similarity can be viewed as a coherent score for an entity of type a appearing in the context. The coherent score S_t for type t is computed as follows:

$$S_t = argmax_{a \in A_t} \Delta(P(T|C), P(T|(C+a))) \tag{2.9}$$

where $P(T|X)$ is the topic distribution of a bag of words X; T is a random variable defined over hidden topics; Δ is a similarity function between the two probability distributions. To simplify the computation, we use the vector dot product of the two topic distributions as the similarity function. The type that has the highest coherent score is selected as the final prediction.

2.6 Experiments

In this section, we introduce the datasets, preprocessing, baselines, experimental settings, and present performance and error analysis on the results.

2.6.1 Datasets

To conveniently make a fair comparison between our transfer learning based FGET and the previous methods, we evaluate the proposed FGET method on the following two standard FGET corpora [4]:

FIGER(GOLD) [56] is the first released FGET corpus, which contains Wikipedia articles annotated with FIGER type taxonomy that has 113 types. The training set was automatically annotated by linking entity mentions via anchor links and mapping Freebase types to FIGER types. The test data consists of manually annotated news reports.

[4]Some other datasets are not publicly available.

	FIGER(GOLD)	OntoNotes (GFT)
# types	113	89
Taxonomy depth	2	3
# raw training mentions	2000895	251039
# raw testing mentions	563	8963

TABLE 2.1: Statistics of FGET corpus

OntoNotes [12] dataset consists of sentences from Wall Street Journal (WSJ). The original OntoNotes 5.0 [43] was annotated with their own type taxonomy that has 19 types. Daniel Gillick et al. [12] re-annotated OntoNotes partially (only the named entity mentions in WSJ) with the GFT taxonomy (89 types) automatically using DBpedia Spotlight. The manually annotated test set was also shared. To compare the performance of our method with previous work, we use the GFT OntoNotes annotation.

We use the TREC-4 dataset to train the HMM-LDA topic model. We only use the Financial Times newswire documents since the government reports in the TREC-4 are different from the newswire documents in terms of style.

2.6.2 Preprocessing

We extract all the constituents of the mention phrases in both corpora, and then set the word vectors of OOVs with transfer learning based on sub-word information. To let the SpanBERT model encode the context efficiently, we adjust the context and position of mention phrases. We slide the context window, so that each mention phrase will be located in approximately the centre of a context without segmenting a sentence or clause. The maximum context sequence length after tokenization was set to 128. Parts of some long contexts before some punctuation marks (comma, semicolon or period that appear before the mention phrases) were dropped. The statistics of the corpora after preprocessing are shown in Table 2.1.

The reasons for choosing 128 tokens as maximum context are as follows:

- We use the datasets [5] shared by Shimaoka et al. [49], where most of the contexts are no longer than 128 tokens.
- Pre-trained Transformer [53] based language models usually tokenize sentences into word pieces [55], and the length of context is a hyperparameter selected from 128, 256, 384 or 512 according to limitations of GPUs.

[5] https://github.com/shimaokasonse/NFGEC

Type Labels	Example of Topic Anchors	# of Topic Anchors
Person/artist	*actor, comedian, singer*	74
Person/business	*billionaire, businessman, entrepreneur*	32
Person/political	*activist, diplomat, senator*	36
Person/athlete	*athlete, boxer, cyclist*	71
Organization/company	*automaker, manufacturer, retailer*	67
Organization/government	*bureau, department, whitehouse*	35

TABLE 2.2: Statistics of Topic Anchors

- Considering that RNN performs poor on translation of sentences longer than 60 words [3], sentences of 128 word pieces are long contexts.

For the topic anchors, we first extract seed topic anchors from the Wikipedia dump using the infobox and the "is-a" pattern in the first sentence of each article. Then regular expressions are used to search such seed topic anchors in the TREC-4 corpus, and those anchors that do not appear in the corpus are scrapped. Some other topic anchors are obtained through the patterns of [41]. These topic anchors are manually aligned with the types on FIGER and GFT taxonomies. Most of the extracted topic anchors belong to the *Person, Organization* type, while some types do not have such topic anchors. The statistics of topic anchors are listed in Table 2.2.

2.6.3 Baselines

To test the effectiveness of our transfer learning schemes, we compare our model with several state-of-the-art FGET models. Our baselines include: (i) **NFETC** model [37], an LSTM based model that counteract label noises by retreating to parental coarse-grained types; (ii) **Attentive** model [49], an LSTM based model with attention mechanism; (iii) **AllC** model [2], an LSTM based model using a variant of hinge loss function; (iv) **AFET** model [45], a model that embed handcrafted features into dimensional space; (v) **ERNIE** [64], using ERNIE contextualized representations to generate context features.

We name the variants of our proposed model as follows: (i) **TLFR**: the base model that only use the **T**ransfer **L**earning based **F**eature **R**epresentation of mentions and contexts; (ii) **TLFR-TA**: add the **T**opic **A**nchor based inference module to the base model.

Parameter description	FIGER(GOLD)	OntoNotes (GFT)
MLP classifier hidden size	400	400
MLP classifier dropout rate	0.8	0.9
Inference threshold	0.215	0.225
Mention LSTM hidden size	300	300
SpanBERT	L-12_H-768_A-12	L-12_H-768_A-12
# Topics	200	200
# Syntactic	10	10

TABLE 2.3: Hyperparameter settings

2.6.4 Experimental Setup

We implement our model using the TensorFlow Estimator[6] framework. Each training example was transformed into three vectors: a fixed-length word-id vectors with padding for mention phrase, a 768-d vector for contextual representation by SpanBERT, a fixed-length one-hot label vector. Such transformations of training and test set are saved as tf_record format. We extract the *fast*Text [7] embeddings of words that appear in mention phrases, and then combine it with the transferred word vectors of OOVs. The combined embedding file is used in the joint training of mention feature learning and classification model. We do not update such word embeddings during training.

For the topic model (HMM-LDA), we use the code [7] shared by Baheti et al. [4] to train the topic model on the TREC-4 corpus. The number of topics is set to 200, and the number of syntactic class is set to 10.

2.6.4.1 Hyperparameter Settings

The hyperparameters for our experiments are listed in Table 2.3. We use the *fast*Text crawl-300d-subword embedding. The hidden size of the LSTM for mention is set to 300. We use the Adagrad optimizer [15] with learning rate 5e-4 as our optimization method. The training iterates 4 epochs.

The threshold for the MLP local binary classifiers is set tentatively. The hidden size of local classifiers is set to 400. Based on the preliminary results, different dropout rates are applied on the both corpora.

[6] https://www.tensorflow.org/guide/estimators
[7] https://github.com/abaheti95/HMM-LDA

2.6.4.2 Evaluation Metrics

We adopt the 3 metrics proposed by Xiao Ling and Daniel S Weld [56], which have been used widely in FGET research. FGET is essentially a multi-class multi-label classification problem, thus the three F1 metrics are based on the precision / recall scores that are computed in three different ways. The *Strict Accuracy* is computed considering the predicted labels of each mention are exactly the same as the true labels. The *Loose Macro* is based on the average precision/recall scores computed for each mention. The *Loose Micro* treat the predicted labels and true labels of all mentions as a whole. We implement the computation of such metrics based on the TensorFlow framework, since there is no readily available TensorFlow code for such metrics.

Let t_m denote the golden true type set for mention m, $\hat{t}_m$denote the type set predicted by FGET system, P denote the mentions detected, G denote the mentions of golden truth. The metrics with different granularity are defined as follows [56]:

- *Strict*: The predicted type is considered correct only if the type set is exactly the same as the golden truth.

$$precision = \frac{\left|\{mentions\ whose\ \hat{t}_m = t_m\}\right|}{|\{mentions\ detected\}|} \tag{2.10}$$

$$recall = \frac{\left|\{mentions\ whose\ \hat{t}_m = t_m\}\right|}{|\{mentions\ golden\ truth\}|} \tag{2.11}$$

- *Loose Macro*: The precision and recall scores are computed independently for each mention, and then denote the average over all mentions as overall metrics.

$$precision = \frac{1}{|P|}\sum_{m\in P}\frac{\left|\hat{t}_m \cap t_m\right|}{\hat{t}_m} \tag{2.12}$$

$$recall = \frac{1}{|G|}\sum_{m\in G}\frac{\left|\hat{t}_m \cap t_m\right|}{t_m} \tag{2.13}$$

- *Loose Micro*: The precision and recall scores are measured globally across all mentions.

$$precision = \frac{\sum_{m\in P}\left|\hat{t}_m \cap t_m\right|}{\sum_{m\in P}\left|\hat{t}_m\right|} \tag{2.14}$$

$$recall = \frac{\sum_{m\in G}\left|\hat{t}_m \cap t_m\right|}{\sum_{m\in G}\left|t_m\right|} \tag{2.15}$$

FGET Model	FIGER(GOLD)			OntoNotes (GFT)		
	Strict Acc.	Macro F1	Micro F1	Strict Acc.	Macro F1	Micro F1
AFET[45]	53.3	69.3	66.4	55.1	71.1	64.7
AllC[2]	65.8	81.2	77.4	52.2	68.5	63.3
Attentive[49]	59.68	78.97	75.36	51.74	70.98	64.91
NFETC[37]	68.9	81.9	79.0	60.2	76.4	70.2
ERNIE[64]	57.19	76.51	73.39	-	-	-
TLFR	67.32	77.44	74.72	60.35	72.34	67.51
TLFR-TA	**69.45**	**82.27**	**79.67**	**61.89**	**76.48**	**70.72**

TABLE 2.4: Performance on the FIGER(GOLD) and OntoNotes corpora

2.6.5 Results Comparison and Analysis

The performance of our models and the baselines on the two standard FGET corpora is listed in Table 2.4. The proposed TLFR model retrieves several labels for many mentions, thus the macro precision and micro precision suffered from such naive threshold. But the strict accuracy of the TLFR model is quite competitive, and the strict accuracy on OntoNotes even better than the previous state-of-the-art models.

2.6.5.1 Performance Analysis

The competitive strict accuracy of TLFR on the corpora indicates that the proposed transfer learning schemes for feature learning can provide the simple binary classifier with more efficient feature representations. Such feature representations enable the model to shield noises on about 60% of the testing examples. However, because of the confusing label noises, the model retrieved some negative labels for other testing examples and thus downgraded the Macro F1 and Micro F1.

Our TLFR-TA model achieved the state-of-the-art results on the corpora. Although our topic anchors are primarily of *Person* and *Organization* types, the topic anchor based inference module boosted the performance because most of the testing examples are also of such two types (as listed in Table 2.5). More importantly, our TLFR-TA model is able to retrieve fine-grained level 2 and 3 types (e.g. */Organization/company, /Person/artist/author*). The pre-trained topic model and topic anchors can rescue the model from confusing label noises in the training set without retreating to coarse-grained types.

The numbers of type labels in the testing sets are listed in Table 2.5. Our topic anchors mainly belong to the listed four level 2 types and */Person/business* (but there is paucity of

	FIGER(GOLD)	OntoNotes (GFT)
# testing examples	563	8963
# label */Organization/∼*	90	1386
# label */Organization/company*	28	1108
# label */Person/∼*	28	472
# label */Person/politician*	5	189
# label */Person/artist*	4	134
# label */Person/athlete*	17	1

TABLE 2.5: Statistics of testing examples in FGET corpora

testing examples for this type). Because of the magnitude difference of the number of testing examples in the two corpora, the topic model retrieved far more level 2 labels on OntoNotes, although it achieved a larger margin of improvement on FIGER.

2.6.5.2 Error Analysis

Error Analysis for TLFR The following deficiency of our TLFR model possibly contribute to the errors. (i) Averaging embeddings of character-level n-grams also incorporate some noises, and the more informative suffix patterns are diluted. (ii) The setting of negative training examples for the local binary classifiers also has great influence on the performance. Daniel Gillick et al. [12] experimented with 3 settings of negative examples, and the setting same as ours achieved the poorest performance. The speculation is that such flooded negative examples may amplify the label noises. Thus using more sophisticated strategy for negative examples is a possible way of improving performance.

Error Analysis for the Topic Model One source of errors is that the candidate types returned by typing model do not include the true type. The other one is that some topic anchors are ambiguous. For example, "investor, contractor" are the topic anchors of both */Organization/company* and */Person/business*, "director" is the topic anchor of both */Person/artist/director* and */Person/business*. Moreover, the topic anchors of different types may appear in the same context. For example, there is "he may be a fine lawyer, he is a bad politician" in the training corpus for our topic model, while "lawyer" is a topic anchor for */Person/legal*, and "politician" is a topic anchor for *Person/political_figure*. Our algorithm that selects the most coherent topic anchors can somewhat offset such ambiguity and confusing context, but estimating the topic distributions based solely on a bag of contextual words is sometimes inaccurate.

2.7 Conclusion and Future Work

In this chapter, we proposed a new transfer learning based approach for fine-grained entity typing that contains three transfer learning schemes. Firstly, to avoid on-site learning word vectors of OOVs in mention phrases, we proposed to generate more precise word embeddings for OOvs through transfer learning using sub-word information. Secondly, instead of learning contextual features using LSTM, we proposed to generate contextual representations through transfer learning using a pre-trained bi-directional non-recurrent neural language model. Thirdly, to reduce the influence of label noises without twisting the original labels, we proposed to refine the predicted labels at inference time using a pre-trained topic model. The topic model associates types with topics through the so-called topic-anchors. The experimental results on two standard FGET corpora validated the effectiveness of our transfer learning approach. Compared with previous methods, our method can predict more fine-grained labels and achieve the state-of-the-art performance.

For future work, we intend to further improve and extend our work as follows. The embeddings of OOVs are currently computed by averaging the ebmeddings of their character-grams. We conjecture that more precise embeddings can be obtained by giving prefix and suffix more weights. The settings for negative examples can be further explored. The algorithm for estimating the topic distribution is relatively too simple to get accurate topic distribution. More work can be done to explore alternative ways of getting more accurate topic distribution. Our work can also be extended by incorporating fine-grained semantic type information in other downstream NLP tasks, such as entity linking and coreference resolution. An unsupervised pre-trained topic model can be used to directly provide semantic type information in a manner similar to our method. In addition to the topic model, another possible way of capturing the topic-type relatedness is using the topic-anchors as distant supervision to compute the coherent scores between topic anchors and context.

References

[1] Abbas Ghaddar and Philippe Langlais. Transforming wikipedia into a large-scale fine-grained entity type corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[2] Abhishek, Ashish Anand, and Amit Awekar. Fine-grained entity type classification by jointly learning representations and label embeddings. In *Proceedngs of European Chapter of Association for Computational Linguistics*, pages 797–807, 2017.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *The 3rd International Conference on Learning Representations (ICLR)*, 2015.

[4] Ashutosh Baheti, Alan Ritter, Jiwei Li, and Bill Dolan. Generating more interesting responses in neural conversation models with distributional constraints. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3970–3980, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D18-1431.

[5] Debrup Banerjee, Kazi Islam, Keyi Xue, Gang Mei, Lemin Xiao, Guangfan Zhang, Roger Xu, Cai Lei, Shuiwang Ji, and Jiang Li. A deep transfer learning approach for improved post-traumatic stress disorder diagnosis. *Knowledge and Information Systems*, 60(3): 1693–1724, 2019.

[6] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[7] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[8] Peter F. Brown, Vincent J. Della Pietra, Peter V. deSouza, Jenifer C. Lai, and Robert L. Mercer. Class-based n-gram models of natural language. *Computational Linguistics*, 18 (4):467–480, 1992. URL https://www.aclweb.org/anthology/J92-4003.

[9] Carlos N. Silla and Alex A. Freitas. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 22(1):31–72, 2011.

[10] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*, 2020. URL https://openreview.net/pdf?id=r1xMH1BtvB.

[11] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM, 2008.

[12] Daniel Gillick, Nevena Lazic, Kuzman Ganchev, Jesse Kirchner, and David Huynh. Contextdependent fine-grained entity type tagging. *arXiv preprint arXiv:1412.1820*, 2014.

[13] Hal Daume III and Daniel Marcu. Domain adaptation for statistical classifiers. *Journal of artificial Intelligence research*, 26:101–126, 2006.

[14] Dong Deng, Liping Jing, Jian Yu, Shaolong Sun, and Michael K Ng. Sentiment lexicon construction with hierarchical supervision topic model. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(4):704–718, 2019.

[15] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul): 2121–2159, 2011.

[16] Asif Ekbal, Eva Sourjikova, Anette Frank, and Simone Paolo Ponzetto. Assessing the challenge of fine-grained named entity recognition and classification. In *proceedings of the 2010 Named Entities Workshop*, pages 93–101, 2010.

[17] Eunsol Choi, Omer Levy, Yejin Choi, and Luke Zettlemoyer. Ultra-fine entity typing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 87–96, 2018-07-15.

[18] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[19] Thomas L Griffiths, Mark Steyvers, David M Blei, and Joshua B Tenenbaum. Integrating topics and syntax. In *Advances in neural information processing systems*, pages 537–544, 2005.

[20] Hailong Jin, Lei Hou, Juanzi Li, and Tiansi Dong. Attributed and predictive entity embedding for fine-grained entity typing in knowledge bases. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 282–292, 2018-08-20.

[21] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

[22] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[23] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[24] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. GloVe: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[25] Mingmin Jin, Xin Luo, Huiling Zhu, and Hankz Hankui Zhuo. Combining deep learning and topic modeling for review understanding in context-aware recommendation. In

Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pages 1605–1614, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1145. URL https://www.aclweb.org/anthology/N18-1145.

[26] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77, 2020.

[27] Gil Keren, Sivan Sabato, and Björn Schuller. Analysis of loss functions for fast single-class classification. *Knowledge and Information Systems*, 62(1):337–358, 2020.

[28] Li Dong, Furu Wei, Hong Sun, Ming Zhou, and Ke Xu. A hybrid neural model for type classification of entity mentions. In *Proceedngs of International Joint Conference on Artificial Intelligence*, pages 1243–1249, 2015.

[29] Mingzhe Liu, Mingfu He, Ruili Wang, and Shaoda Li. A new local density and relative distance based spectrum clustering. *Knowledge and Information Systems*, 61(2):965–985, 2019.

[30] Denghao Ma, Yueguo Chen, Kevin Chen-Chuan Chang, Xiaoyong Du, Chuanfei Xu, and Yi Chang. Leveraging fine-grained wikipedia categories for entity search. In *Proceedings of the 2018 World Wide Web Conference*, pages 1623–1632, 2018.

[31] Masatoshi Suzuki, Koji Matsuda, Satoshi Sekine, Naoaki Okazaki, and Kentaro Inui. Fine-grained named entity classification with wikipedia article vectors. In *IEEE/WIC/ACM International Conference on Web Intelligence*, 2016.

[32] Pablo N. Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. DBpedia spotlight: shedding light on the web of documents. In *Proceedings of the 7th international conference on semantic systems*, pages 1–8. ACM, 2011.

[33] Michael Fleischman and Eduard Hovy. Fine grained classification of named entities. In *The 19th International Conference on Computational Linguistics*, 2002.

[34] Mohamed Amir Yosef, Sandro Bauer, Johannes Hoffart, Marc Spaniol, and Gerhard Weikum. HYENA: Hierarchical type classification for entity names. In *Proceedngs of COLING 2012*, pages 1361–1370, 2012.

[35] Arvind Neelakantan and Ming-Wei Chang. Inferring missing entity type instances for knowledge base completion: New dataset and methods. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 515–525, Denver, Colorado, May–June

2015. Association for Computational Linguistics. doi: 10.3115/v1/N15-1054. URL https://www.aclweb.org/anthology/N15-1054.

[36] Nitish Gupta, Sameer Singh, and Dan Roth. Entity linking via joint encoding of types, descriptions, and context. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 2671–2680, 2017-09-07.

[37] Peng Xu and Denilson Barbosa. Neural fine-grained entity type classification with hierarchy-aware loss. In *Proceedings of NAACL-HLT*, pages 16–25, 2018-06-01.

[38] Matthew Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. Semi-supervised sequence tagging with bidirectional language models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1756–1765, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1161. URL https://www.aclweb.org/anthology/P17-1161.

[39] Maxim Rabinovich and Dan Klein. Fine-grained entity typing with high-multiplicity assignments. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 330–334, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-2052. URL https://www.aclweb.org/anthology/P17-2052.

[40] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/research-covers/languageunsupervised/language understanding paper. pdf*, 2018.

[41] Will Radford and James R. Curran. Joint apposition extraction with syntactic and semantic constraints. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 671–677, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P13-2118.

[42] Altaf Rahman and Vincent Ng. Inducing fine-grained semantic classes via hierarchical and collective classification. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 931–939, Beijing, China, August 2010. Coling 2010 Organizing Committee. URL https://www.aclweb.org/anthology/C10-1105.

[43] Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, and Michelle Franchini.

Ontonotes release 5.0 with OntoNotes DB tool v0.999 beta. In *Linguistic Data Consortium*, 2013.

[44] Marta Recasens, Marie-Catherine de Marneffe, and Christopher Potts. The life and death of discourse entities: Identifying singleton mentions. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 627–633, Atlanta, Georgia, June 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/N13-1071.

[45] Xiang Ren, Wenqi He, Meng Qu, Lifu Huang, Heng Ji, and Jiawei Han. AFET: Automatic fine-grained entity typing by hierarchical partial-label embedding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1369–1378, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1144. URL https://www.aclweb.org/anthology/D16-1144.

[46] Xiang Ren, Wenqi He, Meng Qu, Clare R Voss, Heng Ji, and Jiawei Han. Label noise reduction in entity typing by heterogeneous partial-label embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1825–1834, 2016.

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

[48] Sanjeev Karn, Ulli Waltinger, and Hinrich Schutze. End-to-end trainable attentive decoder for hierarchical entity classification. In *Proceedngs of European Chapter of Association for Computational Linguistics*, pages 752–758, 2017.

[49] Sonse Shimaoka, Pontus Stenetorp, Kentaro Inui, and Sebastian Riedel. Neural architectures for fine-grained entity type classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1271–1280, Valencia, Spain, April 2017. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/E17-1119.

[50] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[51] Tomas Mikolov, Greg Corrado, Kai Chen, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *ICLR Worshop*, 2013.

[52] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013.

[53] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[54] Gregor Wiedemann, Eugen Ruppert, Raghav Jindal, and Chris Biemann. Transfer learning from lda to bilstm-cnn for offensive language detection in twitter. In *14th Conference on Natural Language Processing (KONVENS)*, 2018.

[55] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

[56] Xiao Ling and Daniel S Weld. Fine-grained entity recognition. In *Proceedngs of 26th AAAI Conference on Artificial Intelligence*, pages 94–100, 2012.

[57] Yadollah Yaghoobzadeh, Heike Adel, and Hinrich Schutze. Corpus-level fine-grained entity typing. *Journal of Artificial Intelligence Research*, 61:835–862, 2018.

[58] Yadollah Yaghoobzadeh, Heike Adel, and Hinrich Schütze. Noise mitigation for neural entity typing and relation extraction. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1183–1194, Valencia, Spain, April 2017. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/E17-1111.

[59] Zhilin Yang, Ruslan Salakhutdinov, and William W Cohen. Transfer learning for sequence tagging with hierarchical recurrent networks. In *LCLR*, 2017.

[60] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*, 2019.

[61] Dani Yogatama, Daniel Gillick, and Nevena Lazic. Embedding methods for fine grained entity type classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 291–296, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-2048. URL https://www.aclweb.org/anthology/P15-2048.

[62] Yukun Ma, Erik Cambria, and Sa Gao. Label embedding for zero-shot fine-grained named entity typing. In *Proceedngs of the 26th International Conference on Computational Linguistics: Technical Papers*, pages 171–180, 2016.

[63] Daochen Zha and Chenliang Li. Multi-label dataless text classification with topic modeling. *Knowledge and Information Systems*, 61(1):137–160, 2019.

[64] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. ERNIE: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1139. URL https://www.aclweb.org/anthology/P19-1139.

[65] Zhuosheng Zhang, Hai Zhao, Kangwei Ling, Jiangtong Li, Zuchao Li, Shexia He, and Guohong Fu. Effective subword segmentation for text comprehension. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(11):1664–1674, 2019.

Chapter 3

Improving Entity Linking through Semantic Reinforced Entity Embeddings

Entity embeddings, which represent different aspects of each entity with a single vector like word embeddings, are a key component of neural entity linking models. Existing entity embeddings are learned from canonical Wikipedia articles and local contexts surrounding target entities. Such entity embeddings are effective, but too distinctive for linking models to learn contextual commonality. We propose a simple yet effective method, FGS2EE, to inject fine-grained semantic information into entity embeddings to reduce the distinctiveness and facilitate the learning of contextual commonality. FGS2EE first uses the embeddings of semantic type words to generate semantic embeddings, and then combines them with existing entity embeddings through linear aggregation. Extensive experiments show the effectiveness of such embeddings. Based on our entity embeddings, we achieved new state-of-the-art performance on entity linking.

3.1 Introduction

Entity Linking (EL) or Named Entity Disambiguation (NED) is to automatically resolve the ambiguity of entity mentions in natural language by linking them to concrete entities in a

Knowledge Base (KB). For example, in Figure 3.1, mentions "Congress" and "Mr. Mueller" are linked to the corresponding Wikipedia entries, respectively.

FIGURE 3.1: Entity linking with embedded fine-grained semantic types.

Neural entity linking models use local and global scores to rank and select a set of entities for mentions in a document. Entity embeddings are critical for the local and global score functions. But the current entity embeddings [13] encoded too many details of entities, thus are too distinctive for linking models to learn contextual commonality.

We hypothesize that fine-grained semantic types of entities can let the linking models learn contextual commonality about semantic relatedness. For example, *rugby* related documents would have entities of *rugby player* and *rugby team*. If a linking model learns the contextual commonality of *rugby* related entities, it can correctly select entities of similar types using the similar contextual information.

In this chapter, we propose a method FGS2EE to inject fine-grained semantic information into entity embeddings to reduce the distinctiveness and facilitate the learning of contextual

commonality. FGS2EE uses the word embeddings of semantic words that represent the hallmarks of entities (e.g., *writer, carmaker*) to generate semantic embeddings. We find that the training converges faster when using semantic reinforced entity embeddings.

Our proposed FGS2EE consists of four steps: (i) creating a dictionary of fine-grained semantic words; (ii) extracting semantic type words from each entity's Wikipedia article; (iii) generating semantic embedding for each entity; (iv) combining semantic embeddings with existing embeddings through linear aggregation.

The rest of this chapter is organized as follows. We introduce the background in Section 3.2. The related work is reviewed in Section 3.3. We describe our motivation, method and experiments in Section 3.4, Section 3.5-3.6 and Section 3.7, respectively.

3.2 Entity Linking Background

3.2.1 Task Description

Given a knowledge base containing a set of entities E and a text document in which a set of named entity mentions M are identified in advance, the goal of entity linking is to map each entity mention $m \in M$ to its corresponding entity $e \in E$ in the knowledge base [30]. Here, a named entity mention m is a token sequence in text which potentially refers to some named entity and is identified in advance. It is possible that some entity mention in text does not have its corresponding entity record in the given knowledge base. This kind of mentions are named '**unlinkable mentions**' and annotated with a special label NIL.

Entity linking is also nameded Named Entity Disambiguation (NED). This task is challenging due to the the many-to-many ambiguity between surface form mentions and the entities they refer to. In this research, we just focus on entity linking for English language, rather than cross-lingual entity linking [40].

Typically, the task of entity linking is preceded by a named entity recognition stage, during which boundaries of named entities in text are identified. While named entity recognition is not the focus of this research, the technical details of approaches for named entity recognition

task can be found in the survey paper [9]. In addition, there are many publicly available named entity recognition tools, such as Stanford NER[1], OpenNLP[2], and LingPipe[3].

3.2.2 Local Models for Entity Linking

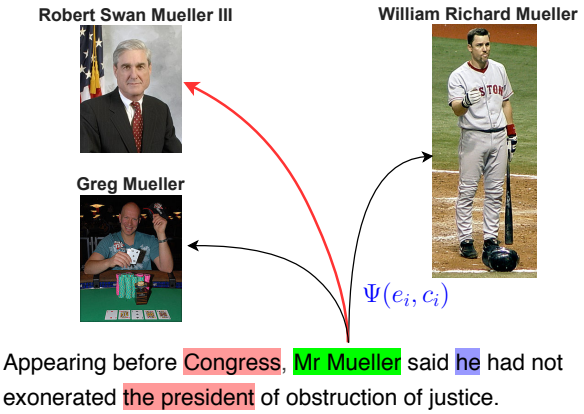

FIGURE 3.2: Local model for entity linking.

Local models rely only on local contexts of mentions and completely ignore interdependencies between the linking decisions in the document (these interdependencies are usually referred to as coherence). Suppose a document D contains a list of mentions $m_1, \ldots, m_n$. Let c_i be a local context of mention m_i and $\Psi(e_i, c_i)$ be a local score function. A local model [21] [38], [13], [22] then tackles the problem by searching the highest scored candidate

$$e_i^* = \underset{e_i \in E_{m_i}}{arg\ max}\ \Psi(e_i, c_i) \tag{3.1}$$

for each mention $i \in \{1, \ldots, n\}$.

The local score $\Psi(e_i, c_i)$ is usually computed as follows:

$$\Psi(e_i, c_i) = \mathbf{e}_i^\top \mathbf{B}\ f(c_i) \tag{3.2}$$

where $\mathbf{e}_i \in \mathbb{R}^d$ is the embedding of entity e_i , $\mathbf{B} \in \mathbb{R}^{d \times d}$ is a diagonal matrix. The mapping $f(c_i)$ applies an attention mechanism to context words in c_i to obtain a feature representations of context ($f(c_i) \in \mathbb{R}^d$).

[1] http://nlp.stanford.edu/ner/
[2] http://opennlp.apache.org/
[3] http://alias-i.com/lingpipe/

Francis-Landau et al. [12] and He et al. [18] use convolutional neural networks (CNNs) and stacked denoising auto-encoders, respectively, to learn representations of textual documents and canonical entity pages. Entities for each mention are locally scored based on cosine similarity with the respective document embedding. In a similar local setting, Sun et al. [31] embed mentions, their immediate contexts and their candidate entities using word embeddings and CNNs. However, their entity representations are restrictively built from entity titles and entity categories only.

Ganea and Hofmann [13] employ attention mechanism [2] to select words that are informative for the disambiguation decision. Their local score is a learned combination of the local entity-mention score and a mention-entity prior $\hat{p}(e_i|m_i)$.

$$\Psi(e_i, m_i, c_i) = f(\Psi(e_i, c_i), \log\ \hat{p}(e_i|m_i)) \tag{3.3}$$

where f is a neural network with two fully connected layers of 100 hidden units and ReLU. The mention-entity prior $\hat{p}(e_i|m_i)$ is computed by averaging probabilities from two indexes build from mention entity hyperlink count statistics from Wikipedia and a large Web corpus. The context-based local score is computed as follows:

$$\Psi(e_i, c_i) = \sum_{w \in \overline{c_i}} \beta(w) \mathbf{e}_i^\top\ \mathbf{B}\ \mathbf{e}_w \tag{3.4}$$

where $\mathbf{e}_w$ is the word embedding of contextual word w. $\overline{c_i}$ is a reduced contextual words set. $\overline{c_i}$ contains only the informative words that are strongly related to at least one candidate entity of mention m_i. $\beta(w)$ is an attention weight for w. Such an attention based local score has a smaller memory footprint and is fast for both training and testing. However, some informative contextual words that not appear in local contexts may be ignored.

3.2.3 Global Models for Entity Linking

A global model, besides using local context with $\Psi(e_i, c_i)$, takes into account entity coherency. It is captured by a coherence score function $\Phi(E, D)$:

$$E^* = \underset{E \in E_{m_1} \times\ \ldots\ \times E_{m_n}}{arg\ max} \sum_{i=1}^{n} \Psi(e_i, c_i) + \Phi(E, D) \tag{3.5}$$

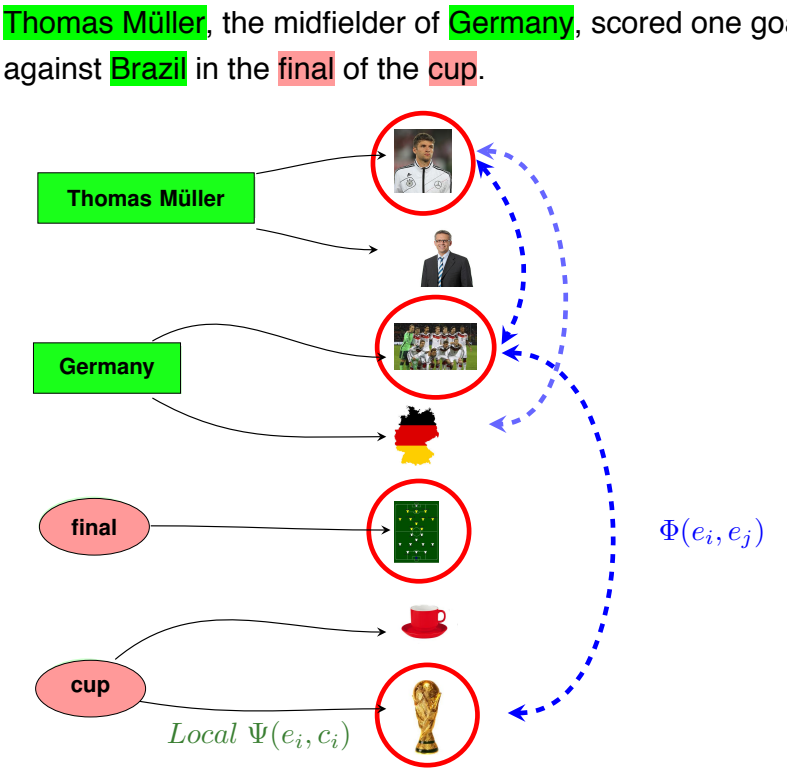

FIGURE 3.3: Global model for entity linking [13].

where $E = (e_1, \ldots, e_n)$. The coherence score function, in the simplest form, is a sum over all pairwise scores $\Phi(e_i, e_j, D)$ ([14], [16], [15], [37], as shown in Figure 3.3), resulting in:

$$E^* = \underset{E \in E_{m_1} \times \ \ldots \ \times E_{m_n}}{arg\ max} \sum_{i=1}^{n} \Psi(e_i, c_i) + \sum_{i \neq j} \Phi(e_i, e_j, D) \tag{3.6}$$

where the pairwise score $\Phi(e_i, e_j, D)$ is usually computed as follows:

$$\Phi(e_i, e_j, D) = \frac{1}{n-1} \mathbf{e}_i^\top \mathbf{C}\ \mathbf{e}_j \tag{3.7}$$

where $\mathbf{e}_i$ and $\mathbf{e}_j \in \mathbb{R}^d$ are are the embeddings of entity e_i, e_j respectively, $\mathbf{C} \in \mathbb{R}^{d \times d}$ is a diagonal matrix. It should be noted that the pairwise score is agnostic to any relations between entities or even to their ordering: it models $e_1, \ldots, e_n$ simply as a bag of entities.

Le and Titov [22] propose to improve the pairwise score by exploiting latent relations between entities. Their pairwise scores take into account relations between mentions which are represented by relation embeddings. They assume that there are K latent relations. Each relation k is assigned to a mention pair (m_i, m_j) with a non-negative weight α_{ijk}. The pairwise score

is computed as a weighted sum of relation-specific pairwise scores.

$$\Phi(e_i, e_j, D) = \sum_{k=1}^{K} \alpha_{ijk} \Phi(e_i, e_j, D) \tag{3.8}$$

They represent each relation k by a diagonal matrix $\mathbf{R}_k \in \mathbb{R}^{d \times d}$, and the pairwise score $\Phi_k(e_i, e_j, D)$ with relation k is computed as follows:

$$\Phi_k(e_i, e_j, D) = \mathbf{e}_i^\top \mathbf{R}_k \ \mathbf{e}_j \tag{3.9}$$

The weights α_{ijk} are normalized scores:

$$\alpha_{ijk} = \frac{1}{Z_{ijk}} \ \exp\left\{ \frac{f(m_i, c_i)^\top \mathbf{D}_k f(m_j, c_j)}{d} \right\} \tag{3.10}$$

where Z_{ijk} is a normalization factor, $f(m_i, c_i)$ is a function mapping (m_i, c_i) onto $\mathbb{R}^d$, and $\mathbf{D}_k \in \mathbb{R}^{d \times d}$ is a diagonal matrix.

A disadvantage of global models is that exact decoding (Equation (3.6)) is NP-hard [33]. Previous work has investigated different approximation techniques, including: random graph walks [16], personalized PageRank [28], inter-mention voting, graph pruning [19], integer linear programming [5], or ranking SVMs [29]. Globerson et al. [15] propose a star model which approximates the decoding problem in Equation (3.6) by approximately decomposing it into n decoding problems, one per each e_i by performing a single round of message passing with attention. Ganea and Hofmann [14], [13] resolve mentions jointly using a fully-connected pairwise conditional random field (CRF) [4] with parametrized potentials. The parameters of potentials are learnt by casting loopy belief propagation (LBP) [27] as a rolled-out deep network. Their linking model directly optimizes the marginal likelihoods, using the same networks for learning and prediction. They use *truncated fitting* of LBP to a fixed number of message passing iterations. This allows for back-propagating through the (truncated) message passing, thereby optimizing the CRF potentials to work well in conjunction with the inference scheme.

3.3 Related Work

Our research focuses on improving the vector representations of entities through fine-grained semantic types. Related topics are as follows.

Entity Embeddings Following the success and ubiquitous application of word embeddings [32], [24], [20] in NLP tasks, works were done to embed entities and words in a common low-dimensional vector space. Similar to word embeddings, entity embeddings are the vector representations of entities. Entity embeddings are a key component to avoid hand-engineered features, multiple disambiguation steps, or the need for additional adhoc heuristics when solving the entity linking task. Entity embedding methods can be categorized into two types: **entity co-occurrences based** and **entity description based**.

The entity co-occurences based methods [37], [41], [11] treat the entity mentions of the same entity as special word. Such methods require data about entity-entity co-occurrences which often suffers from sparsity.

The entity description based method [13], [22] bootstrap entity embeddings from their canonical entity pages and local context of their hyperlink annotations. This allows for more efficient training and alleviates the need to compile co-linking statistics. However, there is discrepancy between the Wikipedia articles, used by this method as training corpus for entity embedding, and the newswire documents used by entity linking and real applications. The information in the canonical entity articles also introduce noises. Ganea and Hofmann [13] learned entity embeddings using words from canonical Wikipedia articles and local context surrounding anchor links. They used Word2Vec vectors [24] of positive words and random negative words as input to the learning objective. Thus their entity embeddings are aligned with the Word2Vec word embeddings.

Fine-grained Entity Typing Fine-grained entity typing is a task of classifying entities into fine-grained types [35] or ultra fine-grained semantic labels [7]. Bhowmik and de Melo [3] used a memory-based network to generate a short description of an entity, e.g. "Roger Federer" is described as 'Swiss tennis player'. In this chapter, we heuristically extract fine-grained semantic types from the first sentence of Wikipedia articles.

Embeddings Aggregation Our research is closely related to the work on aggregation and evaluation of the information content of embeddings from different sources (e.g., polysemous words have multiple sense embeddings), and fusion of multiple data sources [34]. Arora et al. [1] hypothesizes that the global word embedding is a linear combination of its sense embeddings. They showed that senses can be recovered through sparse coding. Mu et al. [26] showed that senses and word embeddings are linearly related and sense sub-spaces tend to intersect over a line. Yaghoobzadeh et al. [36] probe the aggregated word embeddings of polysemous

words for semantic classes. They created a WIKI-PSE corpus, where word and semantic class pairs are annotated using Wikipedia anchor links, e.g., "apple" has two semantic classes: *food* and *organization*. A separate embedding for each semantic class was learned based on the WIKI-PSE corpus. They found that the linearly aggregated embeddings of polysemous words represent well their semantic classes.

The most similar work is that of Gupta et al. [17], but there are many differences: (i) they use the FIGER [35] type taxonomy that contains manually curated 112 types organized into 2 levels; we employ over 3000 vocabulary words as type, and we treat them as a flat list; (ii) they mapped the Freebase types to FIGER types,but this method is less credible, as noted by Daniel Gillick et al. [8]; we extract type words directly from Wikipedia articles, which is more reliable. (iii) their entity vectors and type vectors are learned jointly on a limited corpus. Ours are linear aggregations of existing entity vectors, and word vectors learned from a large corpus, such fine-grained semantic word embeddings are helpful for capturing informative context.

3.4 Motivation

Coarse-grained semantic types (e.g. *person*) have been used for candidate selection [13]. We observe that fine-grained semantic words appear frequently as apposition (e.g., *Defense contractor* Raytheon), coreference (e.g., the *company*) or anonymous mentions (e.g., *American defense firms*). These fine-grained types of entities can help capture local contexts and relations of entities.

Some of these semantic words have been used for learning entity embeddings, but they are diluted by other unimportant or noisy words. We reinforce entity embeddings with such fine-grained semantic types.

3.5 Extracting Fine-grained Semantic Types

We first create a dictionary of fine-grained semantic types, then we extract fine-grained types for each entity.

3.5.1 Semantic Type Dictionary

We select those words that can encode the hallmarks of individual entities. Desiderata are as follows:

- **profession/subject**, e.g., *footballer*, *soprano, biology, rugby.*
- **title**, e.g., *president, ceo, head, director.*
- **industry/genre**, e.g., *carmaker*, *manufacturer, defense contractor*, *hip hop.*
- **geospatial**, e.g., *canada, asian*, *australian.*
- **ideology/religion**, e.g., *communism, buddhism.*
- **miscellaneous**, e.g., *book, film, tv, ship, language.*

We extract noun frequency from the first sentence of each entity in the Wikipedia dump. Then some seed words are manually selected from frequent nouns. We use word similarity to extend these seed words and finally got a dictionary with 3,227 fine-grained semantic words.

Specifically, we use `spaCy`[4] to compute the similarity between words. For each seed word, we find the top 100 similar words that also appear in Wikipedia articles. We then manually select semantic words from these extended words.

3.5.2 Extracting Semantic Types

For each entity, we extract at most 11 dictionary words (phrases) from its Wikipedia article. For example, "Robert Mueller" in Figure 3.1 will be typed as [*american, lawyer, government, official, director*]. The reasons for selecting at most 11 types are as follows: (i) most entities have about 10 types; (ii) too few types cannot inject effective information; (iii) too many types may introduce noisy words that are not directly related to the entity.

3.5.3 Remapping Semantic Words

For some semantic words (e.g., *conchologist*) or semantic phrases (e.g., *rugby league*), there are no word embeddings available for generating the semantic entity embeddings. We remap these semantic words to semantically similar words that are more common. For example, the *conchologist* is remapped to *zoologist*, and *rugby league* is remapped to *rugby_league.*

[4] https://spacy.io/

3.6 FGS2EE: Injecting Fine-Grained Semantic Information into Entity Embeddings

FGS2EE first uses semantic words of each entity to generate semantic entity embeddings, then combine them with existing entity embeddings to generate semantic reinforced entity embeddings.

3.6.1 Semantic Entity Embeddings

Based on the semantic words of each entity, we can produce a semantic entity embedding. We treat each semantic word as a sense of an entity. The embedding of each sense is represented by the Word2Vec embedding of the semantic word. Suppose we only consider T semantic words for each entity, and the set of entity words of entity e is denoted as S_e. Then the semantic entity embedding $\mathbf{e}^s$ of entity e is generated as follows:

$$\mathbf{e}^s = \frac{1}{T}\sum_{i=1}^{T}\mathbf{e}_{w_i} \tag{3.11}$$

where $w_i \in S_e$ is the ith semantic word, $\mathbf{e}_{w_i}$ is the Word2Vec embedding[5] of semantic word w_i. If $|S_e| < T$, then $T = |S_e|$.

3.6.2 Semantic Reinforced Entity Embeddings

We create a semantic reinforced embedding for each entity by linearly aggregating the semantic entity embeddings and Word2Vec style entity embeddings [13] (hereafter referred to as "Wikitext entity embeddings").

Our semantic entity embeddings tend to be homogeneous. If we average them with the Wikitext embeddings, the aggregated embeddings would be homogeneous too. Thus the entity linking model would not be able to distinguish between those similar candidates. Our semantic reinforced entity embedding is a weighted sum of semantic entity embedding and Wikitext entity embedding, similar to [36]. We use a parameter α to control the weight of semantic entity embeddings. Thus the aggregated (semantic reinforced) entity embeddings achieve a trade-off between homogeneity and heterogeneity.

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

$$\mathbf{e}^a = (1 - \alpha)\ \mathbf{e}^w + \alpha\ \mathbf{e}^s \tag{3.12}$$

where $\mathbf{e}^w$ is the Wikitext entity embedding of entity e.

3.7 Experiments

3.7.1 Datasets

We use the Wikipedia dump 20190401 to extract fine-grained semantic type dictionary and semantic types for entities. We use the Wikitext entity embeddings shared by Le and Titov [22, 23]. For entity linking corpora, we use the datesets shared by Ganea and Hofmann [13] and Le and Titov [22, 23]. The publicly available corpora for entity linking are listed as follows.

AIDA-CoNLL [19] contains AIDA-train for training, AIDA-A for dev, and AIDA-B for testing, having respectively 946, 216, and 231 documents.

MSNBC, AQUAINT, ACE2004, were cleaned and updated by Guo and Barbosa [16], and have respectively 20, 50, and 36 documents for test only.

WNEDCWEB (CWEB), WNED-WIKI (WIKI), were automatically extracted from ClueWeb and Wikipedia [16], [10]. and have 320 documents each for test only.

3.7.2 Evaluation Metrics

Like many other NLP tasks, entity linking is usually evaluated in terms of *precision*, *recall*, F_1 measure, and *accuracy*. The *precision* of an entity linking system is computed as the fraction of correctly linked entity mentions that are generated by the system.

$$precision = \frac{|\{correctly\ linked\ entity\ mentions\}|}{|\{linked\ mentions\ generated\ by\ system\}|} \tag{3.13}$$

The *recall* is computed as the fraction of correctly linked entity mentions that should be linked.

$$recall = \frac{|\{correctly\ linked\ entity\ mentions\}|}{|\{entity\ mentions\ that\ should\ be\ lined\}|} \tag{3.14}$$

Entity Embeddings	Linking Methods	AIDA-B	MSNBC	AQUAINT	ACE2004	CWEB	WIKI	Avg
Wikipedia								
-	Milne and Witten [25]	-	78	85	81	64.1	81.7	77.96
-	Ratinov et al. [29]	-	75	83	82	56.2	67.2	72.68
-	Hoffart et al. [19]	-	79	56	80	58.6	63	67.32
-	Cheng and Roth [5]	-	90	90	86	67.5	73.4	81.38
-	Chisholm and Hachey [6]	84.9	-	-	-	-	-	-
Wiki + Unlabelled documents								
-	Lazic et al. [21]	86.4	-	-	-	-	-	-
Ganea and Hofmann [13]	Le and Titov [23]	**89.66±0.16**	92.2±0.2	90.7±0.2	**88.1±0.0**	78.2±0.2	81.7±0.1	86.18
$T = 6, \alpha = 0.1$	Le and Titov [23]	89.58±0.2	**92.3±0.1**	90.93±0.2	87.88±0.17	**78.47±0.11**	81.71±0.21	86.26
$T = 11, \alpha = 0.2$	Le and Titov [23]	89.23±0.31	92.15±0.24	**91.22±0.18**	88.02±0.15	78.29±0.17	**81.92±0.36**	**86.32**
Wiki + Extra supervision								
-	Chisholm and Hachey [6]	88.7	-	-	-	-	-	-
Fully-supervised(Wiki+ AIDA train)								
-	Guo and Barbosa [16]	89.0	92	87	88	77	**84.5**	85.7
-	Globerson et al. [15]	91.0	-	-	-	-	-	-
Yamada et al. [37]	Yamada et al. [37]	91.5	-	-	-	-	-	-
Ganea and Hofmann [13]	Ganea and Hofmann [13]	92.22±0.14	93.7±0.1	**88.5±0.4**	88.5±0.3	**77.9±0.1**	77.5±0.1	85.22
Ganea and Hofmann [13]	Le and Titov [22]	93.07±0.27	93.9±0.2	88.3±0.6	89.9±0.8	77.5±0.1	78.0±0.1	85.5
Ganea and Hofmann [13]	DCA Yang et al. [39]	**93.73±0.2**	93.80±0.0	88.25±0.4	90.14±0.0	75.59±0.3	**78.84±0.2**	85.32
$T = 6, \alpha = 0.1$	Le and Titov [22]	92.29±0.21	94.1±0.24	88.0±0.38	90.14±0.32	77.23±0.18	77.16±0.43	85.33
$T = 11, \alpha = 0.2$	Le and Titov [22]	92.63±0.14	**94.26±0.17**	88.47±0.23	**90.7±0.28**	77.41±0.21	77.66±0.23	**85.7**

TABLE 3.1: F1 scores on six test sets. The last column is the average of F1 scores on the five out-domain test sets.

F_1 score is defined as the harmonic mean of precision and recall.

$$F_1 = \frac{2 \cdot precision \cdot recall}{precision + recall} \tag{3.15}$$

When the entity mentions that should be linked are give as the input, entity linking systems are usually evaluated using the in-KB *accuracy* and *micro F1* (averaged per mention) [13] metrics.

For a fair comparison with prior work, we use the standard *micro F1* score as evaluation metric. Our data and source code are publicly available at `https://github.com/fhou80/EntEmb/`.

3.7.3 Experimental Settings

The parameters T in Equation (3.11) and α in Equation (3.12) are critical for the effectiveness of our semantic reinforced entity embeddings. We got two sets of entity embeddings with two combinations of parameters: $T = 6, \alpha = 0.1$ and $T = 11, \alpha = 0.2$

To test the effectiveness of our semantic reinforced entity embeddings, we use the entity linking models **mulrel** [22] (ment-norm $K = 3$) and **wnel** [23] that are publicly available. We do not optimize their entity linking code. We just replace the entity embeddings with our semantic reinforced entity embeddings.

Similar to Ganea and Hofmann [13] and Le and Titov [22, 23], we run our system 5 times for each combination of entity embeddings and linking model, and report the mean and 95% confidence interval of the micro F1 score.

3.7.4 Results

The results on six testing datasets are shown in Table 3.1. For the **mulrel** model, our entity embeddings ($T = 11, \alpha = 0.2$) improved performance drastically on MSNBC, ACE2004 and average of out-domain test sets. Be aware that CWEB and WIKI are believed to be less reliable [13]. For the **wnel** model, our both sets of entity embeddings are more effective for four of the five out-domain test sets and the average.

Our entity embeddings are better than that of Ganea and Hofmann [13] when tested on the **mulrel** [22] (ment-norm $K = 3$) and **wnel** [23] entity linking models. Ganea and Hofmann

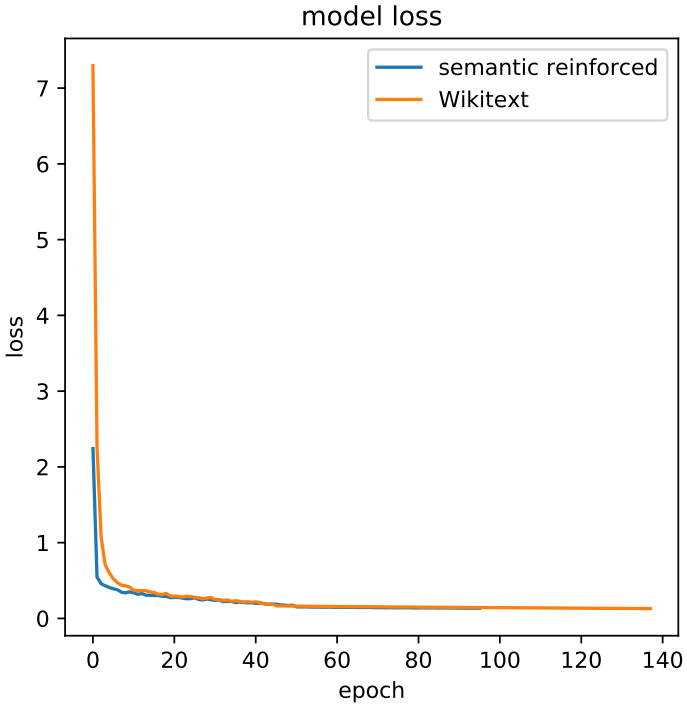

FIGURE 3.4: Learning curves of **mulrel** [22] using two different sets of entity embeddings.

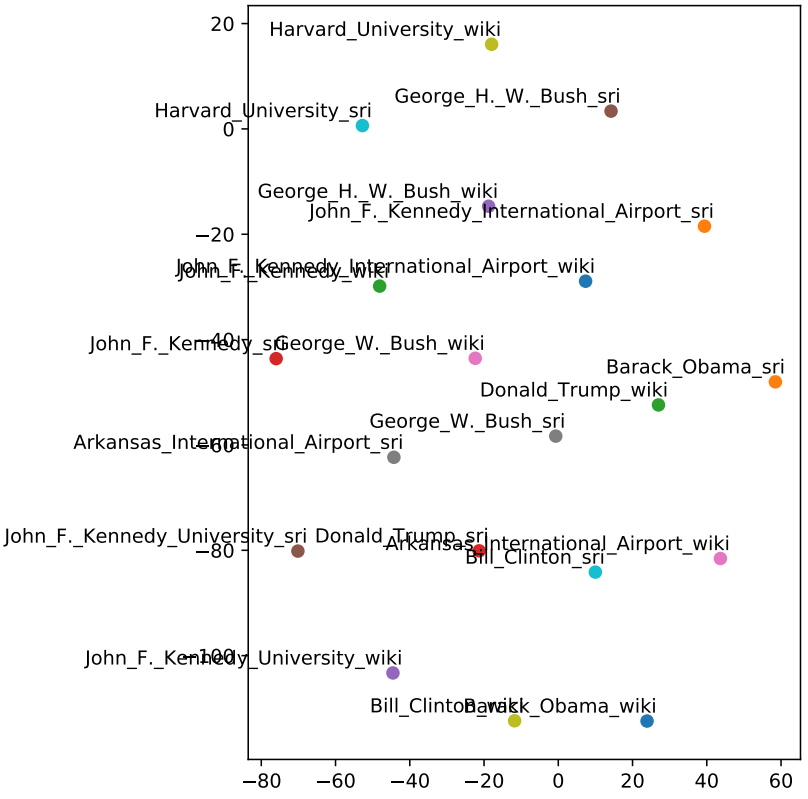

FIGURE 3.5: T-SNE visualization of two sets of entity embeddings. Suffix "_wiki" denotes the Wikitext entity embeddings, while suffix "_sri" denotes the semantic reinforced entity embeddings ($T = 11, \alpha = 0.2$).

[13] showed that their entity embeddings are better than that of Yamada et al. [37] using the entity relatedness metrics.

One notable thing for our semantic reinforced entity embeddings is that the training using our entity embeddings converges much faster than that using Wikitext entity embeddings, as shown in Figure 3.4. One reasonable explanation is that the fine-grained semantic information lets the linking models capture the commonality of semantic relatedness between entities and contexts, hence facilitate the training.

The properties of two different sets of entity embeddings can be visually manifested in Figure 3.5. Our semantic reinforced entity embeddings draw entities of similar types closer, and entities of different types further. For example, our semantic reinforced embeddings of "John F. Kennedy University" and "Harvard University" are closer than the Wikitext embeddings, while our embeddings of "John F. Kennedy International Airport" and "John F. Kennedy" are further. We believe this property contributes to the faster convergence.

3.8 Conclusion

In this chapter, we presented a simple yet effective method, FGS2EE, to inject fine-grained semantic information into entity embeddings to reduce the distinctiveness and facilitate the learning of contextual commonality. FGS2EE first uses the word embeddings of semantic type words to generate semantic embeddings, and then combines them with existing entity embeddings through linear aggregation. Our entity embeddings draw entities of similar types closer, while entities of different types are drawn further. Thus can facilitate the learning of semantic commonalities about entity-context and entity-entity relations. We have achieved new state-of-the-art performance using our entity embeddings.

For the future work, we are planning to extract fine-grained semantic types from unlabelled documents and use the relatedness between the fine-grained types and contexts as distant supervision for entity linking.

References

[1] Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. Linear algebraic structure of word senses, with applications to polysemy. *Transactions of the*

Association for Computational Linguistics, 6:483–495, 2018. doi: 10.1162/tacl_a_00034.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *The 3rd International Conference on Learning Representations (ICLR)*, 2015.

[3] Rajarshi Bhowmik and Gerard de Melo. Generating fine-grained open vocabulary entity type descriptions. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 877–888, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1081. URL https://www.aclweb.org/anthology/P18-1081.

[4] Charles Sutton and Andrew McCallum. An introduction to conditional random fields. *Foundations and Trends in Machine Learning*, 4(4):267–373, 2011. doi: 10.1561/2200000013.

[5] Xiao Cheng and Dan Roth. Relational inference for wikification. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1787–1796, 2013. URL https://www.aclweb.org/anthology/D13-1184.

[6] Andrew Chisholm and Ben Hachey. Entity disambiguation with web links. *Transactions of the Association for Computational Linguistics*, 3:145–156, 2015. doi: 10.1162/tacl_a_00129.

[7] Eunsol Choi, Omer Levy, Yejin Choi, and Luke Zettlemoyer. Ultra-fine entity typing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 87–96. Association for Computational Linguistics, 2018. doi: 10.18653/v1/P18-1009.

[8] Daniel Gillick, Nevena Lazic, Kuzman Ganchev, Jesse Kirchner, and David Huynh. Contextdependent fine-grained entity type tagging. *arXiv preprint arXiv:1412.1820*, 2014.

[9] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007.

[10] Evgeniy Gabrilovich, Michael Ringgaard, and Amarnag Subramanya. FACC1: Freebase annotation of ClueWeb corpora, version 1 (release date 2013-06-26, format version 1, correction level 0). 2013.

[11] Wei Fang, Jianwen Zhang, Dilin Wang, Zheng Chen, and Ming Li. Entity disambiguation by knowledge and text jointly embedding. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 260–269, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/K16-1026. URL https://www.aclweb.org/anthology/K16-1026.

[12] Matthew Francis-Landau, Greg Durrett, and Dan Klein. Capturing semantic similarity for entity linking with convolutional neural networks. *arXiv preprint arXiv:1604.00734*, 2016.

[13] Octavian-Eugen Ganea and Thomas Hofmann. Deep joint entity disambiguation with local neural attention. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2619–2629, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1277. URL https://www.aclweb.org/anthology/D17-1277.

[14] Octavian-Eugen Ganea, Marina Ganea, Aurelien Lucchi, Carsten Eickhoff, and Thomas Hofmann. Probabilistic bag-of-hyperlinks model for entity linking. In *Proceedings of the 25th International Conference on World Wide Web*, pages 927–938. International World Wide Web Conferences Steering Committee, 2016.

[15] Amir Globerson, Nevena Lazic, Soumen Chakrabarti, Amarnag Subramanya, Michael Ringaard, and Fernando Pereira. Collective entity resolution with multi-focal attention. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 621–631, 2016. doi: 10.18653/v1/P16-1059. URL https://www.aclweb.org/anthology/P16-1059.

[16] Zhaochen Guo and Denilson Barbosa. Robust named entity disambiguation with random walks. *Semantic Web (Preprint)*, 9(4):459–479, 2018.

[17] Nitish Gupta, Sameer Singh, and Dan Roth. Entity linking via joint encoding of types, descriptions, and context. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2681–2690, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1284. URL https://www.aclweb.org/anthology/D17-1284.

[18] Zhengyan He, Shujie Liu, Mu Li, Ming Zhou, Longkai Zhang, and Houfeng Wang. Learning entity representation for entity disambiguation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 30–34, 2013.

[19] Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. Robust disambiguation of named entities in text. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 782–792. Association for Computational Linguistics, 2011. URL http://www.aclweb.org/anthology/D11-1072.

[20] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. GloVe: Global vectors

for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[21] Nevena Lazic, Amarnag Subramanya, Michael Ringgaard, and Fernando Pereira. Plato: A selective context model for entity resolution. *Transactions of the Association for Computational Linguistics*, 3:503–515, 2015. doi: 10.1162/tacl_a_00154. URL https://www.aclweb.org/anthology/Q15-1036.

[22] Phong Le and Ivan Titov. Improving entity linking by modeling latent relations between mentions. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1595–1604, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1148. URL https://www.aclweb.org/anthology/P18-1148.

[23] Phong Le and Ivan Titov. Boosting entity linking performance by leveraging unlabeled documents. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1935–1945, Florence, Italy, July 2019. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P19-1187.

[24] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013.

[25] David Milne and Ian H Witten. Learning to link with wikipedia. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 509–518. ACM, 2008.

[26] Jiaqi Mu, Suma Bhat, and Pramod Viswanath. Geometry of polysemy. In *Proceedings of the 5th International Conference on Learning Representations*, 2017.

[27] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[28] Maria Pershina, Yifan He, and Ralph Grishman. Personalized page rank for named entity disambiguation. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 238–243, 2015.

[29] Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. Local and global algorithms for disambiguation to wikipedia. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1375–1384. Association for Computational Linguistics, 2011. URL https://www.aclweb.org/anthology/P11-1138.

[30] Wei Shen, Jianyong Wang, and Jiawei Han. Entity linking with a knowledge base: Issues,

techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering*, 27 (2):443–460, 2014.

[31] Yaming Sun, Lei Lin, Duyu Tang, Nan Yang, Zhenzhou Ji, and Xiaolong Wang. Modeling mention, context and entity with neural networks for entity disambiguation. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[32] Tomas Mikolov, Greg Corrado, Kai Chen, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *ICLR Worshop*, 2013.

[33] Martin J Wainwright, Michael I Jordan, et al. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1–305, 2008.

[34] Ruili Wang, Wanting Ji, Mingzhe Liu, Xun Wang, Jian Weng, Song Deng, Suying Gao, and Chang-an Yuan. Review on mining data from multiple data sources. *Pattern Recognition Letters*, 109:120–128, 2018. doi: 10.1016/j.patrec.2018.01.013.

[35] Xiao Ling and Daniel S Weld. Fine-grained entity recognition. In *Proceedngs of 26th AAAI Conference on Artificial Intelligence*, pages 94–100, 2012.

[36] Yadollah Yaghoobzadeh, Katharina Kann, Timothy J. Hazen, Eneko Agirre, and Hinrich Schütze. Probing for semantic classes: Diagnosing the meaning content of word embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5740–5753, Florence, Italy, July 2019. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P19-1574.

[37] Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. Joint learning of the embedding of words and entities for named entity disambiguation. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 250–259, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/K16-1025. URL https://www.aclweb.org/anthology/K16-1025.

[38] Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. Learning distributed representations of texts and entities from knowledge base. *Transactions of the Association for Computational Linguistics*, 5:397–411, 2017.

[39] Xiyuan Yang, Xiaotao Gu, Sheng Lin, Siliang Tang, Yueting Zhuang, Fei Wu, Zhigang Chen, Guoping Hu, and Xiang Ren. Learning dynamic context augmentation for global entity linking. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 271–281, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1026. URL

https://www.aclweb.org/anthology/D19-1026.

[40] Tao Zhang, Kang Liu, and Jun Zhao. Cross lingual entity linking with bilingual topic model. In *Twenty-Third IJCAI*, 2013.

[41] Stefan Zwicklbauer, Christin Seifert, and Michael Granitzer. Robust and collective entity disambiguation through semantic embeddings. In *Proceedings of the 39th International ACM SIGIR conference*, pages 425–434. ACM, 2016.

Chapter 4

Improving Entity Linking through Anonymous Entity Mentions

State-of-the-art named entity linking models normally use both local and global contextual information for ranking candidates. Global information exploits document level coherence of the referenced entities by computing a pair-wise score between candidates of a pair of named entity mentions (e.g., Raytheon and Boeing) in a document. However, in a document, named entity mentions are significantly less frequent than anonymous entity mentions (e.g., defense contractor and the company). In this chapter, we propose a method, DOC-AET, to exploit the coherence between candidate entities and anonymous entity mentions in a DOCument. We use the Anonymous Entity Type (AET) words to extract anonymous entity mentions. We learn embeddings of AET words from the AET words' inter-paragraph co-occurrence matrix, thus the document-level entity type relatedness is encoded in the AET word embeddings. Then, we build AET entity embeddings and document AET context embeddings using the AET word embeddings. The coherence scores between candidate entities and anonymous entities are computed using the AET entity embeddings and document context embeddings. By incorporating such coherence scores for candidates ranking, DOC-AET has achieved new state-of-the-art results on three of the five out-domain test sets for named entity linking.

4.1 Introduction

Named Entity Linking (NEL) or Named Entity Disambiguation (NED) is the task of linking the ambiguous entity mentions in textual documents to the corresponding entities in a Knowledge

Base (KB). For example, in Figure 3.1, the referenced entity of the mention "Nardelli" should be the *chief executive* "Robert Nardelli". NEL has been used in pre-processing for tasks such as information extraction [16] and question answering [33].

Entity linking systems typically consist of two sequential modules: candidate entities generation and candidate entities ranking [27]. Research on NEL has largely focused on two types of contextual information for candidate entities ranking: local information and global information. Local information is based on words that appear in the context window around an entity mention. For global information, the document-level coherence of the referenced entities is exploited to make compatible linking decisions collectively [12, 31]. The global coherence score is a pair-wise score computed by a bilinear form of the entity embeddings of candidates of a pair of entity mentions in a document [10]. Multiple latent relations between mentions in a document are also exploited to capture coherence [20]. Another way of using global information is to sequentially link and accumulate dynamic context information from linked entities [32].

All the aforementioned methods of using global information exploit the information of candidate entities of named entity mentions (e.g., "Nardelli" and "Home Depot Inc"). However, such named entity mentions appear less frequently than anonymous entity mentions (e.g., *the company*[1] in Figure 4.1). Thus, such methods can only use limited global information, but the more frequently occurring anonymous entity mentions are ignored. The anonymous entity mentions always appear as fine-grained entity type words (e.g., the *company, Canadian singer, service provider, news agency* etc.). These words are parts of anonymous entity mentions, and we call such words **Anonymous Entity Type** (**AET**) words.

Coarse-grained entity type information (e.g., *person, organization, location*) has been used for candidate entities selection [7, 10]. Fine-grained entity type information from BERT-encoding [3] or Wikipedia articles [17] was incorporated into entity representations for candidates ranking. However, these efforts focus on the type information of named entity mentions.

Our hypothesis is that the type information of anonymous entity mentions can capture more contextual information. We can use the anonymous entity mentions in a document to infer the types of the named entity mentions. For example, in Figure 3.1, *company* and *chief executive* are highly related with each other in documents; when ranking the candidate entities

[1] We use italic font to represent anonymous entity mentions

of "Nardelli", the entity "Robert Nardelli" with type *chief executive* is more coherent with the document that has many anonymous *company* mentions.

In this chapter, we propose a method DOC-AET to exploit DOCument-level coherence of named entity mentions and anonymous entity mentions for improving entity linking. We use the 3,227 fine-grained type words of Hou et al. [17] [2] as AET vocabulary. We use AET words to extract anonymous entity mentions. We first learn embeddings of AET words from the document-level AET words inter-paragraph co-occurrence matrix. Then we build AET entity embeddings and document AET context embeddings using the AET word embeddings. The coherence scores between candidate entities and anonymous entities are computed using the AET entity embeddings and document context embeddings. By incorporating such coherence scores for candidate ranking, we achieved new state-of-the-art performance on three of the five out-domain test sets for NEL.

Our contributions can be summarized as follows:

- We are the first to explore the document-level relatedness of fine-grained types. We propose a novel method to capture the relatedness of AET words from document-level context, i.e., extract AET words' inter-paragraph co-occurrence and learn AET word embeddings. The document-level relatedness of AET words is encoded in the AET word embeddings.
- We incorporate a new coherence score based on AET entity embeddings and document's AET context embeddings.
- We verify the effectiveness of the incorporated coherence score on standard benchmark datasets and achieve significant improvement over the baselines.

The rest of this chapter is organized as follows. We introduce background in Section 4.2. The related work is reviewed in Section 4.3. The overview of our method is provided in Section 4.4. We describe our method and experiments in Section 4.5-4.6 and Section 4.7 respectively.

[2] https://github.com/fhou80/EntEmb/

4.2 Background

4.2.1 Named Entity Linking

Formally, given a knowledge base (KB) that contains a set of entities E and a document D in which a set of named entity mentions M are identified in advance, the goal of entity linking is to link each entity mention $m_i \in M$ to its corresponding entity $e_i \in E$. It is possible that an entity mention does not have its corresponding entity in the given KB (i.e., e_i =NIL).

Because $|E|$ can be very large, entity linking systems typically consist of two modules: candidate entity generation and candidate entity ranking. Candidate entity generation is to select possibly referenced entities E_m in the KB for mention m. Candidate entity ranking is to rank the candidate entities in E_m to find out which entity $e \in E_m$ is the most likely referenced entity. Research on NEL has largely focused on the following two types of candidate ranking scores.

4.2.2 Local Score for Candidate Ranking

The local context score $\Psi(e_i, c_i)$ measures the relevance of entity candidates of each mention independently. Neural network based NEL models usually compute Ψ as follows:

$$\Psi(e_i, c_i) = \mathbf{e}_i^\top \mathbf{B}\ f(c_i) \tag{4.1}$$

where $\mathbf{e}_i \in \mathbb{R}^d$ is the embedding of candidate entity e_i; $\mathbf{B} \in \mathbb{R}^{d \times d}$ is a diagonal matrix; $f(c_i) \in \mathbb{R}^d$ is a feature representation of local context c_i surrounding mention m_i.

The local context score is combined with the context-independent mention-entity prior $\hat{p}(e|m)$ [10] as follows:

$$\Psi(e_i, c_i, m_i) = f(\Psi(e_i, c_i), \hat{p}(e_i|m_i)) \tag{4.2}$$

where f is a neural network with two fully connected layers and ReLU activation function.

4.2.3 Global Score for Candidate Ranking

The global score adds a pairwise score $\Phi(e_i, e_j, D)$ to take the coherence between entities in document D into account.

$$\Phi(e_i, e_j, D) = \frac{1}{n-1} \mathbf{e}_i^\top \mathbf{C}\ \mathbf{e}_j \tag{4.3}$$

where $\mathbf{e}_i$ and $\mathbf{e}_j \in \mathbb{R}^d$ are the embeddings of entity e_i, e_j, which are candidates for mention m_i and m_j, respectively; $\mathbf{C} \in \mathbb{R}^{d \times d}$ is a diagonal matrix. The pairwise score of Le and Titov [20] considers K latent relations between entities.

$$\Phi(e_i, e_j, D) = \sum_{k=1}^{K} \alpha_{ijk}\ \mathbf{e}_i^{\top} \mathbf{R}_k\ \mathbf{e}_j \tag{4.4}$$

where α_{ijk} is the weight for relation k, and $\mathbf{R}_k$ is a diagonal matrix for measuring relations k between two entities.

4.3 Related Work

Our research focuses on improving NEL by exploiting coherence of candidate entities' type and anonymous entities' type. We use linear aggregations of AET word embeddings to build AET candidate entity embeddings and AET document context embeddings. Related topics are as follows.

4.3.1 NEL Using Entity Type Information

Coarse-grained entity type information (e.g., *person, organization, location*) has been used for candidate entities selection [7, 10]. Fine-grained entity type information is usually encoded into entity embeddings. Entity embeddings are the vector representations of entities built from entity-entity co-occurrences [9, 31, 35], or canonical Wikipedia articles and local context surrounding anchor links [10].

Gupta et al. [14] map entities' Freebase types to the FIGER [28] types, and learn entity embeddings and type embeddings jointly on the training data. Chen et al. [3] extract latent entity type information from the embeddings generated by applying the pre-trained BERT encoder to the Wikipedia context of entities. The FGS2EE method [17] injects entity type information into entity embeddings directly. FGS2EE first gets fine-grained types of entities by extracting type words from the first paragraph of Wikipedia articles, and then computs typed entity embeddings by averaging the Word2Vec vectors of type words. The linear aggregations of Ganea and Hofmann [10] entity embeddings and typed entity embeddings are used for candidate ranking.

These efforts focus on the type information of named entity mentions. As such, we aim to exploit the coherence between candidate entities and anonymous entity types (mentions) in the documents.

4.3.2 Word Embeddings

Word embeddings, such as Word2Vec [22] and Glove [25], exclusively exploit the intra-sentence context of words to capture the semantic and syntactic similarities. In this chapter, we use the inter-paragraph co-occurrence of AET words to capture the document-level relatedness of anonymous entity types.

4.3.3 Embeddings Aggregation

It has been proven that the global word embedding is a linear combination of its sense embeddings [1, 24]. Global embeddings of polysemous words [30] or entities [17] can be obtained by linear aggregations. We use linear aggregations of AET word embeddings to generate AET based representations of entities and a document. More details about related work on embeddings aggregation can be found in Section 3.3 of Chapter 3.

4.3.4 Fine-grained Entity Typing

To use entity type information, the entities must firstly be typed. Fine-grained entity typing (FGET) is a task of classifying entities into fine-grained types [28] or ultra fine-grained semantic labels [6]. Mention-level FGET only infers the entity types that are coherent with a specific context [11], while entity-level FGET considers all possible types [29]. Gupta et al. [14] mapped the Freebase types of entities to FIGER [28] types, but this method is less credible, as noted by Gillick et al. [11]. Bhowmik and de Melo [2] used a memory-based network to generate a short description of an entity, e.g., "Roger Federer" is described as 'Swiss tennis player'. Hou et al. [17] heuristically extract fine-grained semantic types from the first paragraph of Wikipedia articles, they employ over 3,000 vocabulary words as type, and treat types as a flat list.

4.4 DOC-AET: Method Overview

4.4.1 Motivation

Our DOC-AET method aims at exploiting the coherence between anonymous entity mentions and candidate entities' types to improve NEL. We observe that fine-grained AET words appear frequently as apposition (e.g., *Defense contractor* Raytheon), coreference (e.g., the *company*) or anonymous entities (e.g., *American defense firms*). We can use the AET words from unlabelled documents to capture the document-level relatedness of anonymous entity types. But to capture the longer contexts and document-level relatedness, we only consider the **inter-paragraph co-occurrence**.

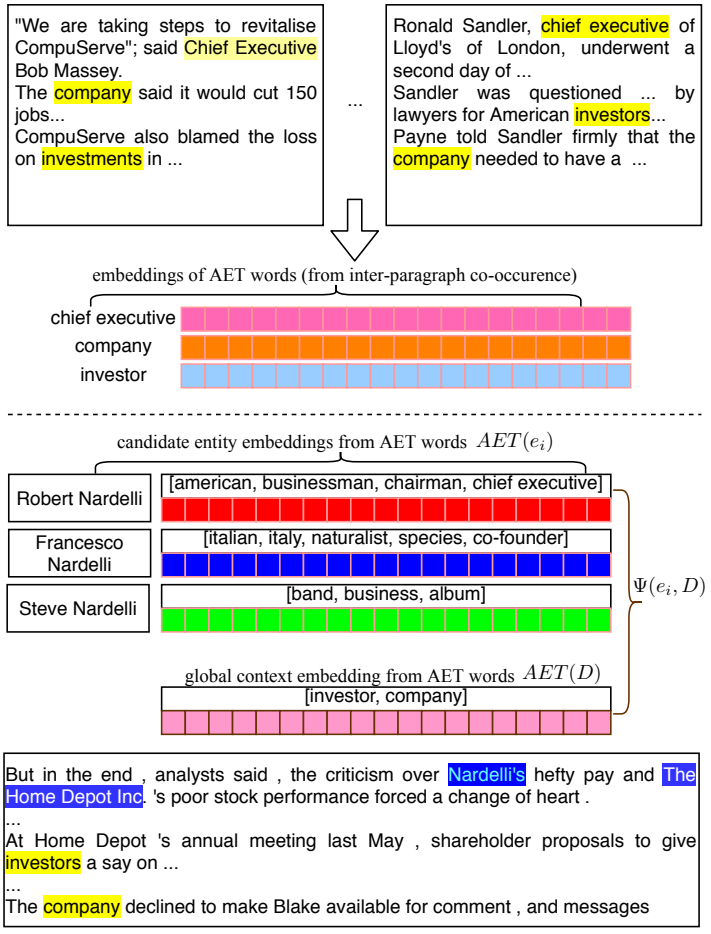

FIGURE 4.1: The process of incorporating the coherence score between entity candidates and Anonymous Entity Type (AET) words (anonymous entity mentions). The AET words are highlighted.

4.4.2 AET Dictionary

We use the type words dictionary of Hou et al. [17] as our AET word dictionary. The dictionary contains 3,227 bigram and unigram types as follows:

- profession/subject, e.g., *footballer*, *soprano, biology, rugby.*
- title, e.g., *president, ceo, head, director.*
- industry/genre, e.g., *carmaker*, *manufacturer, defense contractor*, *hip hop.*
- geospatial, e.g., *canada, asian*, *australian.*
- ideology/religion, e.g., *communism, buddhism.*
- miscellaneous, e.g., *book, film, tv, ship, language.*

4.4.3 Process of DOC-AET Method

Exploiting the coherence between anonymous entity mentions and candidate entities' types is not trivial. As shown in Figure 4.1, the general process can be summarized as follows:

Step 1: Extract anonymous entity mentions (highlighted words) from unlabelled documents; build document-level inter-paragraph co-occurrence matrix; learn inter-paragraph AET words embeddings from co-occurrence. This step is shown in the upper part above the dashed line.

Step 2: Generate AET entity embeddings using the entity types shared by [17]. For example, the "Steve Nardelli" has three types: *band*, *business* and *album*, the entity embedding is generated by averaging the embeddings of these three AET words.

Step 3: Incorporate a coherence score $\Psi(e_i, D)$ between candidate entities' embeddings and document AET context embeddings. For example, the document has two AET words: *investor* and *company*, the AET context embedding is generated by averaging the embeddings of these two AET words.

4.5 Generate AET Word Embeddings

We build AET words' inter-paragraph co-occurrence matrix from unlabelled documents and then learn the word embeddings from the inter-paragraph co-occurrence matrix. This process is similar to method of Glove [25].

4.5.1 Document-level Inter-paragraph Co-occurrence of AET Words

The local context score in Equation (4.1) captures the local context within a sentence, while our DOC-AET score aims at capturing the entity type relatedness across paragraphs (i.e. longer context). We only extract inter-paragraph co-occurrence of AET words, instead of the immediate neighbouring context words.

For each document, we extract a list of AET words from each paragraph. Each document is transformed into a structure of two-dimensional list of AET words. For example, the document in Figure 4.2 can be represented as: [[*'online'*, *'service'*], [], [*'chief'*, *'executive'*], [*'company'*, *'programme'*]].

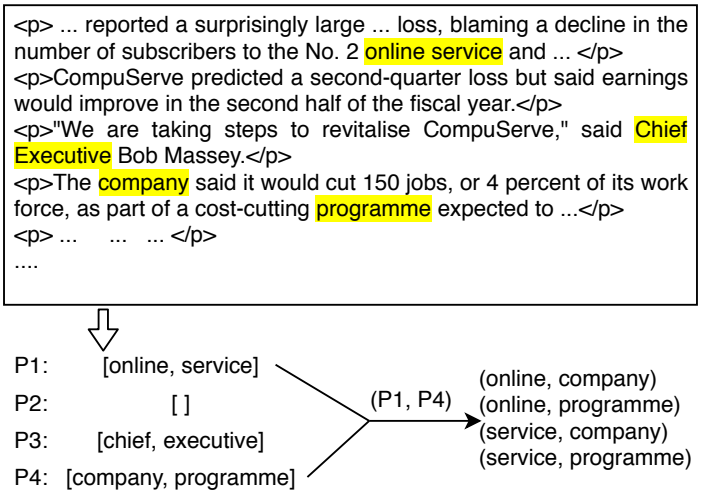

FIGURE 4.2: Building AET words inter-paragraph co-occurrence matrix. AET words are highlighted.

As shown in Figure 4.2, we build AET words' inter-paragraph co-occurrence matrix from the structured representations of documents. Each paragraph is treated as a word. For a pair of paragraphs within context window, we pick one AET word from the left paragraph and right paragraph respectively to count co-occurrence. For example, if paragraph [*'online'*, *'service'*] and paragraph [*'company'*, *'programme'*] are picked, the co-occurrences of (*online*,*company*), (*online*, *programme*), (*service*,*company*) and (*service*,*programme*) are updated. Paragraph pairs that are p paragraphs apart contribute $1/p$ to the total count. We build a symmetric co-occurrence matrix.

4.5.2 Learn AET Word Embeddings

We use the weighted least squares regression model of Glove [25] to learn the AET word embeddings. The cost function is as follows:

$$J = \sum_{i,j=1}^{V} h(X_{ij})(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij})$$

where X_{ij} is the co-occurrence count of word i and j, $w \in \mathbb{R}^a$ are the AET word embeddings. The weighting function $h(x)$ is defined as follows:

$$h(x) = \begin{cases} (x/x_{max})^\alpha, & \text{if } x < x_{max} \\ 1, & \text{otherwise} \end{cases} \tag{4.5}$$

The model generates two sets of word vectors w and $\tilde{w}$, $\tilde{w} \in \mathbb{R}^a$ are separate word embeddings (w and $\tilde{w}$ are equivalent as our X is symmetric) .

4.6 Incorporating AET Scores

4.6.1 Entity Embeddings from AET Words

We use the entity types shared by [17]. Suppose entity e has T AET words, the AET entity embedding $\mathbf{a}_e$ of e is generated by averaging the AET word embeddings of these T words.

$$\mathbf{a}_e = \frac{1}{T} \sum_{i=1}^{T} w_i \tag{4.6}$$

where $w \in \mathbb{R}^a$ are the AET word embeddings.

4.6.2 Document Context Embeddings from AET Words

The document AET context embedding $\mathbf{a}_D$ is generated similarly by averaging the embeddings of AET words extracted from the document. Suppose L AET words are extracted from document D, the AET context embeddings of D is

$$\mathbf{a}_D = \frac{1}{L} \sum_{i=1}^{L} w_i$$

4.6.3 Local AET Scores Using Document Context

The AET coherence score of entity e_i is computed as follow:

$$\Psi(e_i, D) = \mathbf{a}_i^\top \mathbf{A}\ \mathbf{a}_D \tag{4.7}$$

where $\mathbf{a}_i \in \mathbb{R}^a$ is the AET embedding of candidate entity e_i; $\mathbf{A} \in \mathbb{R}^{a \times a}$ is a diagonal matrix; $\mathbf{a}_D \in \mathbb{R}^a$ is the AET context embedding of document D. After incorporating this score, Equation (4.2) becomes:

$$\Psi(e_i, c_i, m_i, D) = f(\Psi(e_i, c_i), \Psi(e_i, D), \hat{p}(e_i|m_i)) \tag{4.8}$$

4.6.4 Model Training

Following Le and Titov [20], we use Equation (4.8) and Equation (4.4) to define a conditional random field (CRF) as follows:

$$q(E_D|D) \propto \Big\{ \sum_{i=1}^{n} \Psi(e_i, c_i, m_i, D) + \sum_{i \neq j} \Phi(e_i, e_j, D) \Big\} \tag{4.9}$$

The max-marginal probability for each mention-candidate is estimated using max-product loopy belief propagation (LBP):

$$\hat{q}_i(e_i|D) \approx \max_{\substack{e_1,\ldots,e_{i-1},\\ e_{i+1},\ldots,e_n}} q(E_D|D) \tag{4.10}$$

The final score for ranking entity candidates is defined as follows:

$$\rho_i(e) = g(\hat{q}_i(e|D), \hat{p}(e, m_i))$$

where g is a two-layer neural network, and $\hat{p}(e|m)$ is the context-independent mention-entity prior.

The other parts of training the model are the same as [20]. The key aspects are as follows:

- The model is trained by minimizing the marginal ranking loss as follows:

$$L(\theta) = \sum_{D \in \mathcal{D}} \sum_{m_i \in D} \sum_{e \in E_{m_i}} max(0, \gamma - \rho_i(e_i^*) + \rho(e)) \tag{4.11}$$

where θ are the model parameters, $\mathcal{D}$ is the collection of training documents.

- To encourage diversity, a regularization term is added to the loss function in Equation 4.11.
- Adam [18] is used as an optimizer.

4.7 Experiments

4.7.1 Datasets for AET Word Embeddings

We use the RCV1, TREC-Disk5 (LA TIMES) and TREC-Disk4 (FINANTIAL TIMES) as training corpus for learning AET word embeddings. These datasets have paragraph segments and our method extracts AET word inter-paragraph co-occurrence from these paragraph segments. We obtain 1,072,120 documents, and 3,140 AET words appear in these documents.

4.7.2 Datasets for NEL

We validate the effectiveness of our method on the following benchmark datasets:

We use the **AIDA-CoNLL** [15] dataset for in-domain training and validation. AIDA-CoNLL contains AIDA-train for training, AIDA-A for dev, and AIDA-B for testing, having respectively 946, 216, and 231 documents.

We evaluate our trained NEL model based on the following out-domain datasets. **MSNBC, AQUAINT, ACE2004**, were cleaned and updated by Guo and Barbosa [13], and have respectively 20, 50, and 36 documents for test. **WNEDCWEB (CWEB), WNED-WIKI (WIKI)**, were automatically extracted from ClueWeb and Wikipedia [8, 13], and both have 320 documents for testing.

Following previous works, we use the pre-processed data shared by Ganea and Hofmann [10] and Le and Titov [20, 21], and consider only mentions that have entities in the KB.

4.7.3 Evaluation Metrics and Baselines

We use the standard Micro-F1 score as evaluation metric. The method of computing **Recall**, **Precision** and **Micro-F1** can be found in the survey of Shen et al. [27].

Our research is following the works of Ganea and Hofmann [10], Le and Titov [20] and Hou et al. [17]. Thus, we use their linking methods (named **DeepEd**, **DeepEd+MulRel**, **DeepEd+MulRel+FGS2EE**, respectively) as baselines. We also compare our method with other state-of-the-art entity linking models.

4.7.4 Experimental Settings

For AET word embeddings, we set the dimension a to 100, and the x_{max}, α in Equation (4.5) are set to 100 and 0.75, respectively. The context window for building inter-paragraph co-occurrence is set to 10. We use the Wikipedia entity typing data shared by [17][3] to generate the AET entity embeddings in Equation (4.6).

In Equation (4.1) and Equation (4.4), we use the semantic reinforced entity embeddings shared by Hou et al. [17].

We modify the PyTorch code of **MulRel** [20] [4] to incorporate the AET coherence score. Following Le and Titov [20], we use the following parameter values: $\gamma = 0.01$ (in Equation 4.11), the number of LBP loops is 10, the f in Equation 4.7 is a neural network with two fully connected layers of 100 hidden units and ReLU non-linearities. We select **ment-norm**, $K = 3$ (in Equation 4.4). The learning rate starts with 10^{-4} and change to 10^{-5} when the F1 score on dev set reaches 91.5%. The model is trained and evaluated on a single GTX 1080 GPU.

Similar to Ganea and Hofmann [10] and Le and Titov [20, 21], we run our NEL system 5 times on the same datasets, and report the mean and 95% confidence interval of the Micro-F1 score.

Our data, source code and trained model are publicly available at `https://github.com/fhou80/DOC-AET`.

4.7.5 Results

The results on six test sets are shown in Table 4.1. The linking methods are categorized into four types.

[3] download from `https://drive.google.com/open?id=1OtLnrH4SpDzdNNcuca-DdXCMwsDPsG3B`
[4] `https://github.com/lephong/mulrel-nel`

Linking Methods	AIDA-B	MSNBC	AQUAINT	ACE2004	CWEB	WIKI
Wikipedia						
Milne and Witten [23]	-	78	85	81	64.1	81.7
Ratinov et al. [26]	-	75	83	82	56.2	67.2
Hoffart et al. [15]	-	79	56	80	58.6	63
Cheng and Roth [4]	-	90	90	86	67.5	73.4
Chisholm and Hachey [5]	84.9	-	-	-	-	-
Wiki + Unlabelled data						
Lazic et al. [19]	86.4	-	-	-	-	-
Le and Titov [21]	89.66±0.16	92.2±0.2	**90.7±0.2**	88.1±0.0	78.2±0.2	81.7±0.1
Wiki + Extra supervision						
Chisholm and Hachey [5]	88.7	-	-	-	-	-
Fully-supervised(Wiki+ AIDA train)						
Guo and Barbosa [13]	89.0	92	87	88	77	**84.5**
Globerson et al. [12]	91.0	-	-	-	-	-
Yamada et al. [31]	91.5	-	-	-	-	-
RMA [34]	91.5	93.2	88.3	89.3	**79.3**	82.2
DeepEd [10]	92.22±0.14	93.7±0.1	88.5±0.4	88.5±0.3	77.9±0.1	77.5±0.1
DeepEd + MulRel [20]	93.07±0.27	93.9±0.2	88.3±0.6	89.9±0.8	77.5±0.1	78.0±0.1
DCA-RL [32]	**93.73±0.2**	93.80±0.0	88.25±0.4	90.14±0.0	75.59±0.3	78.84±0.2
DeepEd + MulRel + FGS2EE [17]	92.63±0.14	94.26±0.17	88.47±0.23	90.7±0.28	77.41±0.21	77.66±0.23
+ DOC-AET	92.59±0.17	**94.55±0.11**	**88.96±0.41**	**91.27±0.14**	77.56±0.14	77.75±0.24

TABLE 4.1: F1 scores on six test sets. The AIDA-B dataset is the in-domain test set, while the other five datasets are out-domain test sets. The methods in bold are our direct baselines.

Firstly, we compare our system to fully-supervised systems, which were trained on AIDA-CoNLL documents. Recall that every mention in these documents has been manually annotated or validated by a human expert. Comparing with all the fully-supervised systems, including our direct baselines **DeepEd** [10], **DeepEd+MulRel** [20] and **DeepEd+MulRel+FGS2EE** [17], our approach is very effective, and achieved significant improvement on three of the five out-domain test sets. The three out-domain test sets, **MSNBC**, **AQUAINT** and **ACE2004** are small data sets manually cleaned and labelled from news articles. The writing styles of these news articles are similar to our datasets for learning AET word embeddings. Comparing with **DeepEd+MulRel+FGS2EE** [17], it is fair to say that incorporating the AET coherence scores can improve performance on all out-domain test sets with slight drop on the in-domain test set.

We then compare our system to the systems that relied on Wikipedia and those which used Wikipedia along with unlabeled data ('Wikipedia + unlabelled data'). Our model outperformed all of them on the in-domain test set and two of the five out-domain test sets. It is seen that the method of Le and Titov [21] outperformed our model on three out-domain test sets: **AQUAINT**, **CWEB** and **WIKI**. But it should be noted that **CWEB** and **WIKI** are believed to be less reliable [10], as they are automatically extracted (all entity linking systems perform comparatively poor on the both test sets). Moreover, their model is trained on a large training set with 30,000 documents, while the AIDA-CoNLL training set only has 946 documents. Our method only extracts the inter-paragraph co-occurrence of 3,140 words.

Linking Methods	**Average F1**
DeepEd [10]	85.22
+MulRel [20]	85.5
+FGS2EE [17]	85.7
+ DOC-AET	**86.02**

TABLE 4.2: Ablation analysis on the effectiveness of our proposed AET coherence scores. The "Average F1" denotes the averaged F1 on the five out-domain test sets.

4.7.6 Ablation Analysis

As we mentioned, our research can be seen as novel but along the line of research by Ganea and Hofmann [10], Le and Titov [20] and Hou et al. [17]. Thus, we perform ablation analysis to gauge contributions of our research.

The method of **DeepEd** [10] is the first to leverage learned neural representations instead of manually designed features. Their deep learning architecture for NEL combines entity embeddings, a neural attention mechanism over local context windows, and unrolled differentiable message passing for global inference. The **MulRel** method [20] improved **DeepEd** by modelling latent multiple relations between textual mentions, i.e., the coherence scores of entity candidates are computed using Equation 4.4 instead of Equation 4.3. Hou et al. [17] inject fine-grained semantic information into entity embeddings to facilitate the learning of contextual commonality.

We use the average F1 score on the five out-domain test sets to conduct ablation comparison, as listed in Table 4.2. The **MulRel** improved the average F1 on the five out-domain test sets by +0.28. Using the semantic reinforced entity embeddings of **FGS2EE** boost the average F1 by +0.2. Incorporating the coherence score between entity candidates and anonymous entity mentions improved the average F1 by +0.32.

4.7.7 Model Complexity

Comparing with the model of **MulRel** [20], our model added the following 200 parameters: (1) 100 parameters are the diagonal matrix $\mathbf{A}$ in Equation 4.7; (2) 100 more parameters are integrated in the f function in Equation 4.8 to incorporate the AET coherence score $\Psi(e_i, D)$.

Thus the complexity of our model should be slightly more expensive than **MulRel** [20] and **DeepEd** [10]. However, our model converges faster than **MulRel**: on average our model needs 80 epochs, while **MulRel** needs 120 epochs and **DeepEd** needs 1250 epochs. In terms of wall-clock time, our model requires less than 1 hour to train on a single GTX 1080, and

AET Words	**AET Cosine Similarity**	**Glove Cosine Similarity**	**Word2Vec Cosine Similarity**
investment	0.9381	0.7935	0.6319
stock	0.9341	0.4737	0.4529
trading	0.9236	0.5374	0.3381
equity	0.9230	0.7325	0.5259
market	0.9098	0.5695	0.4209
finance	0.8875	0.5504	0.3015
fund	0.8851	0.6151	0.3248
bank	0.8829	0.5119	0.2584
portfolio	0.8826	0.5637	0.3864
firm	0.8816	0.4944	0.3218

TABLE 4.3: Cosine similarity between "*investor*" and other AET words using different embeddings

the difference in training time between our model and **MulRel** is negligible. The training of **MulRel** is ten times faster than that of **DeepEd**.

4.7.8 AET Word Embeddings Evaluation

Our AET word embeddings are learnt from AET words' inter-paragraph co-occurrence, thus they can capture the related anonymous entity mentions from longer contexts that may span several paragraphs. Such AET word embeddings are used to compute the coherence scores between entity candidates and other anonymous entity mentions in the same document.

In contrast, the Glove [25] and Word2Vec embeddings are learnt from local context. Such embeddings can only be used to compute the coherence between entity candidates and local context (Equation 4.1).

To demonstrate the difference between our AET word embeddings and Glove/Word2Vec, we list the cosine similarities between *investor* and other AET words using different word embeddings in Table 4.3. The words in the left column are the top-10 most similar words of *investor* using AET embeddings. The documents they appear in are similar to the documents where *investor* appears, thus the AET cosine similarities are higher. In contrast, the local contexts where they appear are different; thus, the Glove and Word2Vec cosine similarities are smaller.

4.8 Conclusion

In this chapter, we present a method DOC-AET to exploit coherence of named entity mentions and anonymous entity type (AET) words/mentions for improving named entity linking. We show that incorporating the coherence score between candidate entities and AET mentions can significantly improve NEL performance. DOC-AET used the fine-grained type words of Hou et al. [17] as AET vocabulary to extract anonymous entity mentions. The document-level relatedness between entity types is encoded into the AET word embeddings which are learnt from the AET words' inter-paragraph co-occurrence matrix. AET entity embeddings and document AET context embeddings are computed using the AET word embeddings. The coherence scores between candidate entities and anonymous entities are computed using the AET entity embeddings and document context embeddings. By incorporating such coherence scores for candidate ranking, we achieve state-of-the-art performance on three of the five out-domain datasets.

For the future work, we plan to apply the document-level entity type coherence to the task of fine-grained entity typing.

References

[1] Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. Linear algebraic structure of word senses, with applications to polysemy. *Transactions of the Association for Computational Linguistics*, 6:483–495, 2018. doi: 10.1162/tacl_a_00034.

[2] Rajarshi Bhowmik and Gerard de Melo. Generating fine-grained open vocabulary entity type descriptions. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 877–888, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1081. URL https://www.aclweb.org/anthology/P18-1081.

[3] Shuang Chen, Jinpeng Wang, Feng Jiang, and Chin-Yew Lin. Improving entity linking by modeling latent entity type information. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, pages 7529–7537, 2020. doi: https://doi.org/10.1609/aaai.v34i05.6251.

[4] Xiao Cheng and Dan Roth. Relational inference for wikification. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1787–1796, 2013. URL https://www.aclweb.org/anthology/D13-1184.

[5] Andrew Chisholm and Ben Hachey. Entity disambiguation with web links. *Transactions of the Association for Computational Linguistics*, 3:145–156, 2015. doi: 10.1162/tacl_a_00129.

[6] Eunsol Choi, Omer Levy, Yejin Choi, and Luke Zettlemoyer. Ultra-fine entity typing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 87–96. Association for Computational Linguistics, 2018. doi: 10.18653/v1/P18-1009.

[7] Greg Durrett and Dan Klein. A joint model for entity analysis: Coreference, typing, and linking. volume 2, pages 477–490. MIT Press, 2014.

[8] Evgeniy Gabrilovich, Michael Ringgaard, and Amarnag Subramanya. FACC1: Freebase annotation of ClueWeb corpora, version 1 (release date 2013-06-26, format version 1, correction level 0). 2013.

[9] Wei Fang, Jianwen Zhang, Dilin Wang, Zheng Chen, and Ming Li. Entity disambiguation by knowledge and text jointly embedding. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 260–269, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/K16-1026. URL https://www.aclweb.org/anthology/K16-1026.

[10] Octavian-Eugen Ganea and Thomas Hofmann. Deep joint entity disambiguation with local neural attention. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2619–2629, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1277. URL https://www.aclweb.org/anthology/D17-1277.

[11] Daniel Gillick, Nevena Lazic, Kuzman Ganchev, Jesse Kirchner, and David Huynh. Contextdependent fine-grained entity type tagging. *arXiv preprint arXiv:1412.1820*, 2014.

[12] Amir Globerson, Nevena Lazic, Soumen Chakrabarti, Amarnag Subramanya, Michael Ringaard, and Fernando Pereira. Collective entity resolution with multi-focal attention. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 621–631, 2016. doi: 10.18653/v1/P16-1059. URL https://www.aclweb.org/anthology/P16-1059.

[13] Zhaochen Guo and Denilson Barbosa. Robust named entity disambiguation with random walks. *Semantic Web (Preprint)*, 9(4):459–479, 2018.

[14] Nitish Gupta, Sameer Singh, and Dan Roth. Entity linking via joint encoding of types, descriptions, and context. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2681–2690, Copenhagen, Denmark, September 2017.

Association for Computational Linguistics. doi: 10.18653/v1/D17-1284. URL https://www.aclweb.org/anthology/D17-1284.

[15] Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. Robust disambiguation of named entities in text. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 782–792. Association for Computational Linguistics, 2011. URL http://www.aclweb.org/anthology/D11-1072.

[16] Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S. Weld. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 541–550, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P11-1055.

[17] Feng Hou, Ruili Wang, Jun He, and Yi Zhou. Improving entity linking through semantic reinforced entity embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6843–6848, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.612. URL https://www.aclweb.org/anthology/2020.acl-main.612.

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[19] Nevena Lazic, Amarnag Subramanya, Michael Ringgaard, and Fernando Pereira. Plato: A selective context model for entity resolution. *Transactions of the Association for Computational Linguistics*, 3:503–515, 2015. doi: 10.1162/tacl_a_00154. URL https://www.aclweb.org/anthology/Q15-1036.

[20] Phong Le and Ivan Titov. Improving entity linking by modeling latent relations between mentions. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1595–1604, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1148. URL https://www.aclweb.org/anthology/P18-1148.

[21] Phong Le and Ivan Titov. Boosting entity linking performance by leveraging unlabeled documents. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1935–1945, Florence, Italy, July 2019. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P19-1187.

[22] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed

representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013.

[23] David Milne and Ian H Witten. Learning to link with wikipedia. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 509–518. ACM, 2008.

[24] Jiaqi Mu, Suma Bhat, and Pramod Viswanath. Geometry of polysemy. In *Proceedings of the 5th International Conference on Learning Representations*, 2017.

[25] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1162. URL https://www.aclweb.org/anthology/D14-1162.

[26] Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. Local and global algorithms for disambiguation to wikipedia. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1375–1384. Association for Computational Linguistics, 2011. URL https://www.aclweb.org/anthology/P11-1138.

[27] Wei Shen, Jianyong Wang, and Jiawei Han. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering*, 27 (2):443–460, 2014.

[28] Xiao Ling and Daniel S Weld. Fine-grained entity recognition. In *Proceedngs of 26th AAAI Conference on Artificial Intelligence*, pages 94–100, 2012.

[29] Yadollah Yaghoobzadeh, Heike Adel, and Hinrich Schutze. Corpus-level fine-grained entity typing. *Journal of Artificial Intelligence Research*, 61:835–862, 2018.

[30] Yadollah Yaghoobzadeh, Katharina Kann, Timothy J. Hazen, Eneko Agirre, and Hinrich Schütze. Probing for semantic classes: Diagnosing the meaning content of word embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5740–5753, Florence, Italy, July 2019. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P19-1574.

[31] Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. Joint learning of the embedding of words and entities for named entity disambiguation. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 250–259, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/K16-1025. URL https://www.aclweb.org/anthology/K16-1025.

[32] Xiyuan Yang, Xiaotao Gu, Sheng Lin, Siliang Tang, Yueting Zhuang, Fei Wu, Zhigang Chen, Guoping Hu, and Xiang Ren. Learning dynamic context augmentation for global entity linking. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 271–281, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1026. URL https://www.aclweb.org/anthology/D19-1026.

[33] Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. Semantic parsing via staged query graph generation: Question answering with knowledge base. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1321–1331, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1128. URL https://www.aclweb.org/anthology/P15-1128.

[34] Xiaoling Zhou, Yukai Miao, Wei Wang, and Jianbin Qin. A recurrent model for collective entity linking with adaptive features. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, volume 34, pages 329–336, 2020.

[35] Stefan Zwicklbauer, Christin Seifert, and Michael Granitzer. Robust and collective entity disambiguation through semantic embeddings. In *Proceedings of the 39th International ACM SIGIR conference*, pages 425–434. ACM, 2016.

Chapter 5

Exploiting Less Anisotropic Span Representations for Entity Coreference Resolution

*It was found that less anisotropic (anisotropic embeddings are not directionally uniform) static word embeddings gain large improvements on downstream NLP tasks. The state-of-the-art coreference resolution models use the output layer of a contextualization model to build span embeddings, and employ the document-level semantics to refine the span embeddings as higher-order coreference resolution. However, the contextualized and higher-order refined span embeddings tend to be highly anisotropic, and make it difficult to distinguish between related but distinct entities (e.g., pilots and flight attendants). In this chapter, we propose five LASE (**L**ess **A**nisotropic **S**pan **E**mbeddings) schemes for coreference resolution, and investigate their effectiveness with experiments. We find that when our **Internal** and **LowerDep** schemes are applied to ELECTRA and SpanBERT, their performances are improved by +1.9 F1 and +0.5 F1 on the OntoNotes benchmark, respectively. Extensive ablation studies also show that the longer-context-encoded contextualized representations of ELECTRA and SpanBERT are more effective than higher-order span embeddings for coreference resolution.*

5.1 Introduction

Coreference resolution is the task of identifying and clustering mentions in a document that refer to the same entity. Traditional coreference resolution models [8, 9, 11, 26, 32] work

in a pipeline fashion. They usually process the task in two stages: mention detection and coreference resolution. At both stages, they rely on syntactic parsers to build complicated hand-engineered features. Such pipelined models suffer from cascading errors and are difficult to be generalized to new datasets and languages [19].

Lee et al. [19] proposed the first end-to-end model that tackles mention detection and coreference resolution simultaneously. They consider all spans as mention candidates, and use two scoring functions to learn which spans are entity mentions and which are their coreferential antecedents. The training objective is to optimize the marginal log-likelihood of all correct antecedents implied by the gold clustering. To control the model complexity, they use a unary mention scoring function to prune the space of spans and antecedents, and a pair-wise antecedent scoring function to compute the softmax distribution over antecedents for each span. Both of the scoring functions are simple feed-forward neural networks, and the input to both scoring functions are the learned span embeddings.

Thus, the core of end-to-end neural coreference resolution models is the learning of span embeddings. Bi-directional LSTM was first used to generate the embedding representations of spans [19, 33]. Following the success of contextualized representations, ELMO [20, 24], BERT [14, 16, 18] and SpanBERT [17] are used to learn span embeddings.

However, the contextualized representations make it difficult to distinguish between related but distinct entities (e.g., *pilots* and *flight attendants* [19]), and have recently been shown to be anisotropic (i.e., not directionally uniform) [12], especially the topmost layer representations. Also, it has been found that less anisotropic embeddings lead to large improvements on downstream NLP tasks [23]. More details about anisotropy can be found in Section **??**.

In this chapter, we propose the following five LASE (**L**ess **A**nisotropic **S**pan **E**mbeddings) schemes to generate less anisotropic span embeddings: Boundary, Internal, Double, Penultimate, LowerDep. Our extensive experiments show that:

1. When our **Internal** and **LowerDep** schemes are applied to ELECTRA [10] and SpanBERT, their performances are improved by +1.9 F1 and +0.5 F1 on the OntoNotes benchmark, respectively.

2. The span embeddings from longer-context-encoded contextualized representations of ELECTRA and SpanBERT are more effective than higher-order span embeddings.

3. The 12th layer embeddings of BERT-base are no better than the 11th layer embeddings for coreference resolution.

4. The degree of anisotropy can be used as guidance for hyperparameter settings.

The rest of this chapter is organized as follows. We introduce background in Section 5.2. The related work is reviewed in Section 5.3. The merits for gauging the degree of anisotropy are given in Section 5.4. The sources of anisotropy are analyzed in Section 5.5. We propose five LASE schemes in Section 5.6. The experimental results are reported and analyzed in Section 5.7.

5.2 Background of End-to-End Neural Coreference Resolution

5.2.1 Task Description

The end-to-end coreference resolution tackles mention detection and coreference resolution simultaneously by span ranking, thus it is formulated as a task of assigning antecedents a_i for each span i. A possible span candidate is any continuous N-gram within a sentence. The set of possible assignments a_i is $\mathcal{A}_i = \{1, \ldots, i-1, \epsilon\}$, ϵ is called a 'dummy' antecedent. If span i is assigned to a non-dummy antecedent – span j, then we have $a_i = j$. If span i is assigned to dummy antecedent ϵ, then it indicates two scenarios: (1) span i is not an entity mention; (2) span i is the first mention of a new entity (cluster). Through transitivity of coreferent antecedents, these assignment decisions induce clusters of entities over the document.

5.2.2 First-order Coreference Resolution

The first-order end-to-end coreference resolution model [19] independently ranks each pair of spans using a pairwise scoring function $s(i, j)$. The scores are then used to compute the antecedent distribution $P(a_i)$ for each span i:

$$P(a_i) = \frac{e^{s(i,a_i)}}{\sum_{j \in \mathcal{A}_i} e^{s(i,j)}} \tag{5.1}$$

The coreferent score $s(i, j)$ for a pair of spans includes three factors: (1) $s_m(i)$, the score for span i being a mention, (2) $s_m(j)$, the score for span j being a mention, (3)$s_a(i, j)$, the score of j being antecedent of i:

$$s(i,j) = \begin{cases} 0 & j = \epsilon \\ s_m(i) + s_m(j) + s_a(i,j) & j \neq \epsilon \end{cases} \tag{5.2}$$

The scoring functions s_m and s_a take span representations $\mathbf{g}$ as input:

$$\begin{aligned} s_m(i) &= \mathbf{w}_m \cdot FFNN_m(\mathbf{g}_i) \\ s_a(i,j) &= \mathbf{w}_a \cdot FFNN_a([\mathbf{g}_i, \mathbf{g}_j, \mathbf{g}_i \circ \mathbf{g}_j, \phi(i,j)]) \end{aligned} \tag{5.3}$$

where $\cdot$ denotes the dot product, $\circ$ denotes element-wise multiplication; $FFNN$ denotes a feed-forward neural network.

However, it is intractable to score every pair of spans in a document. There are $O(W^2)$ spans of potential mentions in a document (W is the number of words). Comparing every pair would be $O(W^4)$ complexity. Thus, pruning is performed according to the mention scores $s_m(i)$ to reduce spans that are unlikely to be an entity mention.

5.2.3 Higher-order Coreference Resolution

The first-order coreference resolution models only consider pairs of spans, and do not directly incorporate any information about the entities to which the spans might belong. Thus, the first-order models may suffer from consistency errors.

Lee et al. [20] proposed a higher-order model that iteratively refine the span representations $\mathbf{g}_i^n$ of span i at nth iteration, using information from antecedents. The refined span representations are used to compute the refined antecedent distribution $P_n(a_i)$:

$$P_n(a_i) = \frac{e^{s(\mathbf{g}_i^n, \mathbf{g}_{a_i}^n)}}{\sum_{j \in \mathcal{A}_i} e^{s(\mathbf{g}_i^n, \mathbf{g}_j^n)}} \tag{5.4}$$

At each iteration, the expected antecedent representation $\mathbf{a}_i^n$ of each span i is computed using the current antecedent distribution $P_n(a_i)$ as an attention mechanism:

$$\mathbf{a}_i^n = \sum_{j \in \mathcal{A}_i} P_n(j) \cdot \mathbf{g}_j^n \tag{5.5}$$

The current span representation $\mathbf{g}_i^n$ is then updated via interpolation with its expected antecedent representation $\mathbf{a}_i^n$:

$$\mathbf{g}_i^{n+1} = \mathbf{f}_i^n \circ \mathbf{g}_i^n + (1 - \mathbf{f}_i^n) \circ \mathbf{a}_i^n \tag{5.6}$$

where $\mathbf{f}_i^n = \sigma(\mathbf{W}_f[\mathbf{g}_i^n, \mathbf{a}_i^n])$ is a learned gate vector. Thus, the span representation $\mathbf{g}_i^{n+1}$ at iteration $n+1$ is an element-wise weighted average of the current span representation $\mathbf{g}_i^n$ and its direct antecedents.

5.2.4 Span Embeddings Based on Contextualized Representations

The core of end-to-end neural coreference resolution models is the learning of the vectorized representations of spans of text. The span representation $\mathbf{g}_i$ of span i is usually a concatenation of four vectors as follow:

$$\mathbf{g}_i = [\mathbf{g}^*_{START(i)}, \mathbf{g}^*_{END(i)}, \hat{\mathbf{g}}_i, \phi(i)] \tag{5.7}$$

where $START(i)$ and $END(i)$ are the start position and end position of span i respectively, **Boundary** representations $\mathbf{g}^*_{START(i)}, \mathbf{g}^*_{END(i)}$ are the vector representations of word $START(i)$ and $END(i)$ respectively. **Internal** representation $\hat{\mathbf{g}}_i$ is a weighted sum of word vectors in span i, $\phi(i)$ is a feature vector encoding the size of span i.

Assume the vector representations of each word are $\{\mathbf{x}_1, \cdots, \mathbf{x}_T\}$ (T is the length of the document), Lee et al. [19], Zhang et al. [33] and Lee et al. [20] use a bi-directional LSTM to build the first three vectors of span representation $\mathbf{g}_i$:

$$\begin{aligned} \mathbf{g}^*_{START(i)} &= BiLSTM(\mathbf{x}_{START(i)}) \\ \mathbf{g}^*_{END(i)} &= BiLSTM(\mathbf{x}_{END(i)}) \\ \hat{\mathbf{g}}_i &= \sum_{t=START(i)}^{END(i)} a_{i,t} \cdot \mathbf{x}_t \end{aligned} \tag{5.8}$$

where $a_{i,t}$ is a learned weight computed from $BiLSTM(\mathbf{x}_t)$, $\mathbf{x}_t$ can be GloVe [15, 19], ELMO [20, 24], or concatenation of GloVe and CNN character embeddings [27, 33].

Following the success of contextualized representations, Kantor and Globerson [18], Joshi et al. [16] replace the LSTM-based encoder with the BERT transformer. They either use BERT in a convolutional mode [18] or split the documents into fixed length before applying BERT [16].

Kantor and Globerson [18] use a learnable weighted average of the **last four layers** of BERT to build span representations. Joshi et al. [16] use the **topmost layer** output of BERT to build span representations:

$$\begin{aligned} \mathbf{g}^*_{START(i)} &= BERT_{l=top}(w_{START(i)}) \\ \mathbf{g}^*_{END(i)} &= BERT_{l=top}(w_{END(i)}) \\ \hat{\mathbf{g}}_i &= \sum_{t=START(i)}^{END(i)} a_{i,t} \cdot BERT_{l=top}(w_t) \end{aligned} \tag{5.9}$$

where $BERT_l(w_i)$ is the lth layer contextualized embeddings of token w_i, $a_{i,t}$ is a learned weight computed from topmost layer output. They split documents into segments of fixed length and apply BERT to each segment. They proposed two variants of splitting: overlap and independent (non-overlapping). Surprisingly, the independent splitting performs better.

5.3 Related Work

End-to-end neural Coreference Resolution Our work on coreference resolution follows the research of Lee et al. [19], Lee et al. [20], Joshi et al. [16], Joshi et al. [17], details about their work can be found in Section 5.2.

Measures of Contextuality Ethayarajh [12] propose to measure how contextual a word representation is using three different metrics: self-similarity, intra-sentence similarity, and maximum explainable variance. We adopt their measures to gauge how anisotropic and how contextual a word representation is, and set hyperparameters of our schemes of exploiting lower layer embeddings.

Embeddings Aggregation Our research is closely related to the work on aggregation and evaluation of the information content of embeddings from different sources (e.g., polysemous words have multiple sense embeddings). More details about related work on embeddings aggregation can be found in Section 3.3 of Chapter 3.

5.4 Gauging Contextualized Representations

Isotropy has both theoretical and empirical benefits. In theory, it allows for stronger "self-normalization" during training [5], and in practice, *less anisotropic* static word embeddings

achieved improvements on several downstream NLP tasks [23]. Anisotropy means that elements of vectorized representations are not uniformly distributed with respect to direction. Instead, they occupy a narrow cone in the vector space. We first gauge the contextualized representations before incorporating lower layer embeddings. We adopt the measures of Ethayarajh [12] to the gauging.

random similarity random similarity is the average cosine similarity between the representations of uniformly randomly sampled words from different contexts. We use it to measure the **degree of anisotropic**.

self-similarity the self-similarity of a word w in layer l is the average cosine similarity between its contextualized representations across its n unique contexts.

intra-sentence similarity intra-sentence similarity of a sentence is the average cosine similarity between its word representations and the sentence vector, which is the mean of those word vectors.

We use self-similarity and intra-sentence similarity to measure the **degree of contextualization**.

5.5 Sources of Anisotropic Span Embeddings

For the higher-order coreference resolution task, we identify the following two sources of anisotropic span embeddings.

- **Higher-order Refining** The span embeddings refinement for higher-order coreference resolution, as shown in Equation (5.5) and (5.6), is essentially injecting information from other spans into a span's embedding. The more iterations for refining the span embeddings, the more anisotropic the span embeddings will be.
- **Contextualized Representations** Ethayarajh [12] show that the contextualized word representations of all words are anisotropic. Representations in higher layers are generally more anisotropic than those in lower ones. Building span embeddings directly from the output layer contextualized word representations will inevitably cause anisotropy.

5.6 Generating Less Anisotropic Span Embeddings

In this section, we describe our LASE schemes of generating less anisotropic span embeddings.

5.6.1 Lower Depth for Higher-order Refinement

To decrease the degree of anisotropy caused by span embedings refinement, we propse the **LowerDep** scheme.

LowerDep scheme: apply span embeddings refinement Equation (5.5) and (5.6) only one iteration.

5.6.2 Using Penultimate Layer Embeddings

For BERT, we gauge that the topmost layer output is no better than the embeddings of the penultimate layer for coreference resolution. Thus we use BERT's penultimate layer embeddings directly for coreference resolution.

Penultimate scheme: use penultimate layer embeddings directly:

$$\begin{aligned} \mathbf{g}^*_{START(i)} &= BERT_{l=penult}(w_{START(i)}) \\ \mathbf{g}^*_{END(i)} &= BERT_{l=penult}(w_{END(i)}) \\ \hat{\mathbf{g}}_i &= \sum_{t=START(i)}^{END(i)} a_{i,t} \cdot BERT_{l=penult}(w_t) \end{aligned} \tag{5.10}$$

5.6.3 Using Linear Aggregations of Multiple Layers Embeddings

For SpanBERT and ELECTRA, we gauge that their contextualized embeddings efficiently encoded the contextual information. Contextualized word representations are more context-specific in higher layers. But the higher the layer, the more anisotropic the contextualized representations. To reduce the degree of anisotropic while retaining the contextual information, we use the linear aggregation of contextualized embeddings of the first layer and the topmost layer. We propose the following three schemes.

Boundary scheme: only apply aggregated embeddings to the boundary representations:

$$\begin{aligned} AT(w_t) &= \alpha \circ T_{l=1}(w_t) + (1-\alpha) \circ T_{l=top}(w_t) \\ \mathbf{g}^*_{START(i)} &= AT(w_{START(i)}) \\ \mathbf{g}^*_{END(i)} &= AT(w_{END(i)}) \end{aligned} \tag{5.11}$$

where $T_l(w)$ is the lth layer Transformer [28] based contextualized embeddings of token w, AT is the aggregated representation, scalar α is the weight of first layer embeddings.

Internal scheme: only apply aggregated embeddings to the internal representation:

$$\begin{aligned} AT(w_t) &= \alpha \circ T_{l=1}(w_t) + (1-\alpha) \circ T_{l=top}(w_t) \\ \mathrm{a}_t &= \mathbf{w}_\mathrm{a} \cdot FFNN_\mathrm{a}(AT(w_t)) \\ a_{i,t} &= \frac{exp(\mathrm{a}_t)}{\sum_{k=START(i)}^{END(i)} exp(\mathrm{a}_k)} \\ \hat{\mathbf{g}}_i &= \sum_{t=START(i)}^{END(i)} a_{i,t} \cdot AT(w_t) \end{aligned} \tag{5.12}$$

Double scheme: apply aggregated embeddings to both the boundary representations and the internal representation.

5.7 Experiments

5.7.1 Implementation and Hyperparameters

We modify the Tensorflow implementation of Transformer based coreference resolution system [1] to incorporate lower layer embeddings. We use the similar hyperparameters, except that we introduce some new hyperparameters: the weight of layer one embeddings α and schemes. The base models are trained on our RTX 2080TI GPUs, while the large models are trained on CPUs.

Similar to [16], we split the OntoNotes English documents into segments of 128, 256, 384, and 512 word pieces and treat the segment length as one hyperparameter. We use cased vocabulary for BERT and SpanBERT, and uncased vocabulary for ELECTRA. We use the

[1] https://github.com/mandarjoshi90/coref

HuggingFace pytorch version ELECTRA [2] (discriminator). We also experiment with lower depth of higher-order coreference resolution.

5.7.2 Baselines

We compare our method with two main baselines: (1) the original c2f-coref + BERT system [16] and (2) c2f-coref + SpanBERT system [17]. We also compare with (3) c2f-coref + ELECTRA system that does not use lower layer contextualized representations to test the effectiveness of lower layer embeddings on ELECTRA.

5.7.3 Data Sets and Evaluation Metrics

5.7.3.1 Document Level Coreference Resolution: OntoNotes

OntoNotes (English) is a document-level dataset from the CoNLL-2012 shared task [25] on coreference resolution. It consists of 2,802/343/348 train/development/test documents of different genres, such as newswire, magazine articles, broadcast news, broadcast conversations etc.

Evaluation Metrics The main evaluation is the average F1 of three metrics: MUC [29], B^3 [6], and $CEAF_{\phi 4}$ [21] on the test set according to the official CoNLL-2012 evaluation scripts. These metrics are based on the comparison between the **key** entities (mentions) and the **response** entities (mentions). The key refers to the gold-standard mentions or entities, while the response denotes the mentions (entities) output by an entity coreference resolution system. The definitions are as follows:

B^3 Metric, a mention-based metric proposed by Bagga and Baldwin [6], computes a precision and recall for each individual mention and takes the average as final metric.

$$Precision = \frac{1}{N}\sum_{i=1}^{N} Precision_{m_i}$$

$$Recall = \frac{1}{N}\sum_{i=1}^{N} Recall_{m_i}$$

where N is the number of mentions in the document.

[2] https://github.com/huggingface/transformers

MUC Metric, proposed during the 6th MUC by Vilain et al. [29], is a link-based metric. MUC score is computed based on the key and response partitions as follows.

$$Precision = \frac{\sum_{i=1}^{N_r}(|R_i| - |\hat{p}(R_i)|)}{\sum_{i=1}^{N_r}(|R_i| - 1)}$$

$$Recall = \frac{\sum_{i=1}^{N_k}(|K_i| - |p(K_i)|)}{\sum_{i=1}^{N_k}(|K_i| - 1)}$$

where R_i is the ith response entity and $\hat{p}(R_i)$ is the set of partitions created by intersecting R_i with kkey entities; K_i is the ith key entity and $p(K_i)$ is the set of partitions created by intersecting K_i with response entities; N_k and N_r are number of the key and response entities. This metric is unable to reward successful indentification of singleton cluster, but is still widely used.

CEAF Metric, proposed by Luo [21], is an entity-based metric. It evaluates coreference outputs based on the best alignment between the clusters in the gold partition and those in the system-generated partition.

CoNLL Metric, an aggregated metric proposed by Pradhan et al. [25] and used by the CoNLL-2012 shared task, is calculated as the average of the B^3 score, MUC score and the CEAF score.

5.7.3.2 Paragraph Level Coreference Resolution: GAP

GAP [30] is a human-labeled corpus of ambiguous pronoun-name pairs derived from Wikipedia snippets. Examples in the GAP dataset fit within a single segment, thus obviating the need for cross-segment inference.

Evaluation Metrics The metrics are F1 score on **M**asculine and **F**eminine examples, **O**verall, and the **B**ias factor (i.e. F/M). Following Webster et al. [30] and Joshi et al. [16], the coreference resolution system is trained on OntoNotes and only the testing is performed on GAP. The dataset and scoring scripts is available at https://github.com/google-research-datasets/gap-coreference.

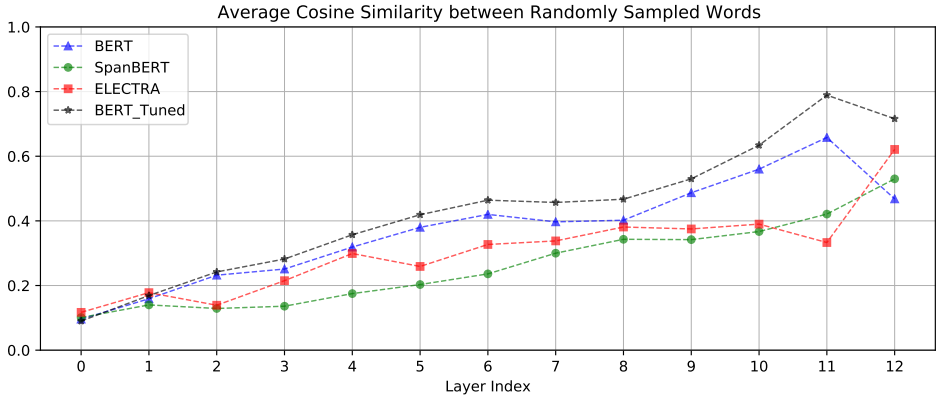

FIGURE 5.1: The degree of anisotropic is measured by random similarity, the average cosine similarity between uniformly randomly sampled words. The higher the layer, the more anisotropic. Embeddings of layer 0 are the input layer word embeddings. (Figure 5.1-5.3 are generated using the method of Ethayarajh [12])

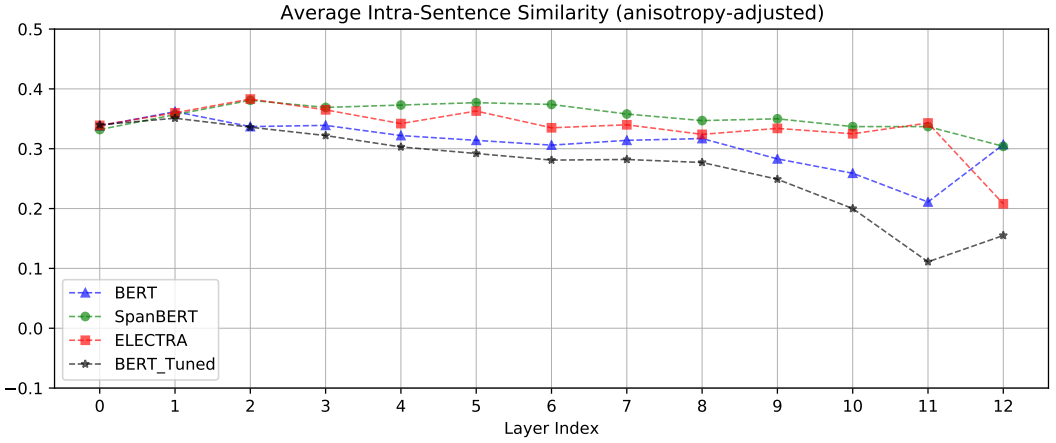

FIGURE 5.2: Intra-sentence similarities of contextualized representations. The intra-sentence similarity is the average cosine similarity between each word representation in a sentence and their mean.

5.7.3.3 Data Sets for Gauging Contextualized Representations

Similar to [12], we use the data from the SemEval Semantic Textual Similarity tasks from years 2012 - 2016 [1–4]. The other settings are also the same.

5.7.4 Results and Findings

In this Subsection, we first report the results on the two data sets, and then we describe our findings and analysis based on the gauging measures of anisotropic and contextualization.

5.7.4.1 Results

The results on the two data sets are listed in Table 5.1 and Table 5.2 respectively. For BERT-base, applying the **Penultimate** scheme achieved the same result as using the topmost layer. However, adding the **LowerDep** scheme to BERT-base causes the performance drop slightly. The results show that the 12th layer embeddings of BERT-base are no better than the 11th layer embeddings for coreference resolution. We can observe that there are abnormal changes from the 11th layer to the 12th layer in Figures 5.1-5.3. We conjecture that the abnormal changes have some connection with the **Penultimate** phenomenon.

For SpanBERT-base, the **Internal+LowerDep** scheme offers an improvement of 0.5% over the original span representations. The improvement over the original ELECTRA-base is 0.9%.

If we see the results in Table 5.1 jointly with Figure 5.1, we find that the more anisotropic, the larger the α value for **Internal** scheme.

The performance of incorporating lower layer embeddings on GAP (Table 5.2) is not as satisfying as that on OntoNotes. From the ablation studies in Table 5.3, we can see that the span representations incorporating lower layer embeddings perform more robust on OntoNotes benchmark, with smaller performance drop from dev set to test set.

For SpanBERT-large and ELECTRA-large, the topmost layer embeddings should be more anisotropic, thus the larger α value is needed. ELECTRA-large+**Internal+LowerDep** scheme achieves the new state-of-the-art performance on the OntoNotes coreference resolution task.

	MUC			B^3			$CEAF_{\phi_4}$			Avg. F1
	P	R	F1	P	R	F1	P	R	F1	
Martschat and Strube [22]	76.7	68.1	72.2	66.1	54.2	59.6	59.5	52.3	55.7	62.5
Clark and Manning [7]	76.1	69.4	72.7	65.6	56.0	60.4	59.4	53.0	56.0	63.0
Wiseman et al. [31]	76.2	69.3	72.6	66.2	55.8	60.5	59.4	54.9	57.1	63.4
Wiseman et al. [32]	77.5	69.8	73.4	66.8	57.0	61.5	62.1	53.9	57.7	64.2
Clark and Manning [9]	79.2	70.4	74.6	69.9	58.0	63.4	63.5	55.5	59.2	65.7
e2e-coref[19]	78.4	73.4	75.8	68.6	61.8	65.0	62.7	59.0	60.8	67.2
c2f-coref[20]	81.4	79.5	80.4	72.2	69.5	70.8	68.2	67.1	67.6	73.0
Fei et al. [13]	85.4	77.9	81.4	77.9	66.4	71.7	70.6	66.3	68.4	73.8
EE[18]	82.6	84.1	83.4	73.3	76.2	74.7	72.4	71.1	71.8	76.6
BERT-base [16]	80.2	82.4	81.3	69.6	73.8	71.6	69.0	68.6	68.8	73.9
+(11th layer, Penultimate)	82.2	80.6	81.4	72.1	71.0	71.5	70.8	66.7	68.7	73.9
SpanBERT-base [17]	84.3	83.1	83.7	76.2	75.3	75.7	74.5	71.2	72.8	77.4
+ ($\alpha = 0.2$, Internal+LowerDep)	84.0	83.7	83.9	**76.2**	76.3	**76.3**	74.5	**72.4**	**73.4**	**77.9**
ELECTRA-base	82.4	**85.1**	83.7	72.0	**78.3**	75.0	73.4	71.5	72.4	77.0
+ ($\alpha = 0.3$, Internal+LowerDep)	**84.3**	84.7	**84.5**	75.1	77.3	76.2	**75.1**	71.2	73.1	**77.9**
SpanBERT-large [17]	85.8	84.8	85.3	78.3	77.9	78.1	76.4	74.2	75.3	79.6
+ ($\alpha = 0.2$, Internal+LowerDep)	**85.8**	85.1	85.4	**78.9**	78.0	78.5	75.8	75.0	75.4	79.8
ELECTRA-large	84.3	85.2	84.7	76.5	78.4	77.4	75.3	73.1	74.2	78.8
+ ($\alpha = 0.4$, Interna+LowerDepl)	85.7	**86.5**	**86.1**	78.2	**80.6**	**79.4**	**77.5**	**75.6**	**76.5**	**80.7**

TABLE 5.1: Results on the test set of the OntoNotes English data from the CoNLL-2012 shared task. The rightmost column is the main evaluation metric, the average F1 of MUC, B^3, $CEAF_{\phi_4}$

.

Model	**M**asculine			**F**eminine			**B**ias factor	**O**verall		
	P	R	F1	P	R	F1		P	R	F1
e2e-coref [19]	-	-	67.2	-	-	62.2	0.92	-	-	64.7
c2f-coref [20]	-	-	75.8	-	-	71.1	0.94	-	-	73.5
BERT-base [16]	-	-	84.4	-	-	81.2	0.96	-	-	82.8
BERT-large [16]	-	-	86.9	-	-	83.0	0.95	-	-	85.0
SpanBERT-base[17]	89.5	**86.1**	**87.7**	85.7	79.5	82.5	0.94	87.6	82.8	85.3
+ ($\alpha = 0.2$, internal)	88.7	85.9	87.3	85.7	**81.8**	**83.7**	**0.96**	87.2	**83.9**	**85.5**
ELECTRA-base	88.8	85.0	86.9	86.9	80.1	83.3	**0.96**	87.9	82.6	85.1
+ ($\alpha = 0.3$, internal)	**90.0**	85.0	87.4	**87.1**	80.0	83.4	0.95	**88.6**	82.5	85.4
SpanBERT-large[17]	92.6	**87.4**	89.9	89.1	80.7	84.7	0.94	90.9	84.0	87.3
+ ($\alpha = 0.2$, internal)	**92.8**	87.3	**90.0**	**89.3**	81.9	85.4	0.95	**91.1**	84.6	87.7
ELECTRA-large	91.9	86.4	89.0	89.0	82.5	85.6	0.96	90.5	84.4	87.3
+ ($\alpha = 0.4$, internal)	92.5	87.0	89.6	89.2	**83.3**	**86.1**	**0.96**	90.8	**85.1**	**87.9**

TABLE 5.2: Performance on the test set of GAP corpus. The metrics are F1 scores on **M**asculine and **F**eminine examples, **O**verall F1 score, and a **B**ias factor(F/M).

	Segment Length Segment Num	Higher-order Depth	α Value	F1 dev	F1 test
SpanBERT-base	384 × 2	2	0	77.4	77.1
		2	0.1	77.7	77.5
		1	0.1	77.9	77.6
	384 × 3	2	0	77.7	**77.4**
		1	0	77.8	77.5
		1	0.1	77.7	77.5
		1	0.2	77.8	**77.9**
	512 × 2	1	0	77.7	77.3
		1	0.2	77.4	77.3
ELECTRA-base	384 × 2	2	0	76.2	76.1
		2	0.2	77.2	77.2
		1	0	76.7	76.9
		1	0.2	77.5	77.5
	384 × 3	1	0	76.9	76.8
		1	0.2	77.5	77.6
		1	0.3	77.8	77.7
	512 × 2	1	0	77.1	**77.0**
		1	0.2	78.0	77.6
		1	0.3	78.0	**77.9**

TABLE 5.3: Ablation studies of ELECTRA-base and SpanBERT-base. The metric is the average F1 score on the OntoNotes dev set and test set using different combinations of hyperparameters. The underlined numbers denote that the performance on the test set is even better than that on the dev set. The bold numbers denote the reported results in Table 5.1.

5.7.4.2 Findings and Analysis

The most effective scheme of incorporating layer 1 embeddings Of the three schemes of incorporating layer 1 embeddings proposed in Section 5.6.3, the most effective one is the **internal** scheme. The **boundary** scheme makes the coreference resolution model perform worse. The **double** scheme performs better but slightly worse than the **internal** scheme. This is in line with the fact that the boundary representations were designed to encode a span's contextual information, while the internal representation was designed to encode the internal information of a span.

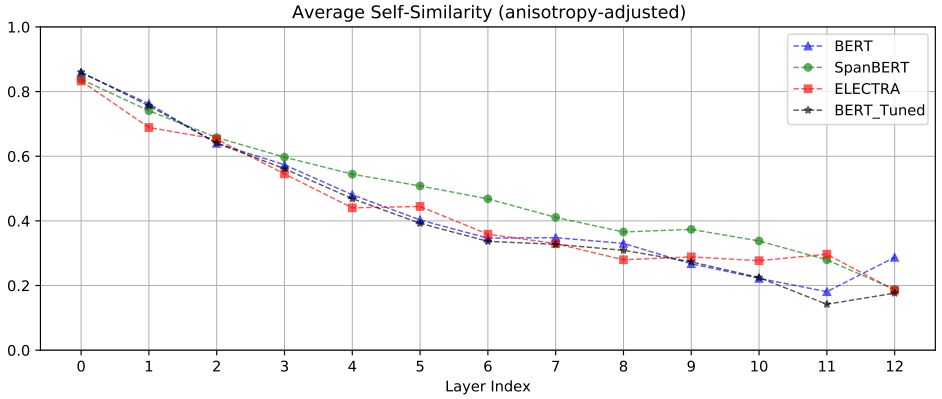

FIGURE 5.3: Self-similarities of contextualized representations. Self-similarity is the average cosine similarity between representations of the same word in different contexts.

Word Embeddings *vs* Layer 1 Embeddings As we can see in Figure 5.1, the layer 0 embeddings (the input layer word embeddings) and layer 1 embeddings are the least anisotropic, thus we also experimented with input layer word embeddings. But we did not achieve salient improvements. The difference is that layer 1 embeddings encode contextual information, thus this shows that the internal representation $\hat{\mathbf{g}}_i$ still needs contextual information to better encode a span's internal structure.

Cased Vocabulary *vs* Uncased Vocabulary

Previous methods [16, 18] use the cased BERT, SpanBERT. Our experiments show that the uncased ELECTRA achieves even better performance.

Higher-order Resolution *vs* Contextualized Representations We performed ablation studies to test the effectiveness of different hyperparameter settings for coreference resolution. As shown in Table 5.3, SpanBERT-base and ELECTRA-base achieve better results when using a lower depth of higher-order coreference resolution. BERT-base still needs a deeper depth of higher-order resolution to get the best result using segments of 128 word pieces.

As mentioned in Section 5.2, higher-order coreference resolution is to incorporate entity-level information. Essentially it is to incorporate contextual information from longer contexts. That is why BERT-base needs higher-order resolution to get the best result using shorter segments. SpanBERT-base and ELECTRA-base achieve best results using segments of 384 and 512 word pieces, respectively. Thus they are capable of encoding longer contextual information. Under this circumstance, incorporating antecedents information using higher-order span embeddings may introduce some misleading features.

This suggests that the span embeddings from longer-context-encoded contextualized representations are better than the higher-order span embeddings for coreference resolution.

Why BERT performs poorly on coreference resolution?

As the results in Table 5.1 show, ELECTRA and SpanBERT gain improvements on OntoNotes benchmark when incorporating layer one embeddings. But we did not achieve any improvement for BERT-base by incorporating layer one embeddings, neither when using the topmost layer embeddings nor when using the penultimate layer embeddings.

As shown in Figure 5.1, contextualized embeddings of BERT are highly anisotropic. Embeddings of every layer are more anisotropic than that of SpanBERT and ELECTRA. The fine-tuned BERT for coreference resolution even becomes more anisotropic.

BERT-base achieve the best performance when using segments of 128 word pieces. This shows that BERT-base is not capable of encoding longer contexts. Figure 5.2 and Figure 5.3 also corroborate that BERT-base does not sufficiently encode the contextual information needed for coreference resolution. This suggests that only the efficiently contextualized representations (e.g. SpanBERT and ELECTRA) benefit from incorporating layer one embeddings.

5.8 Conclusion

In this chapter, we proposed five LASE schemes to generate less *anisotropic* span embeddings for coreference resolution. Before applying LASE, we use three measures to gauge the contextualized representations of BERT, SpanBERT and ELECTRA.

The **Internal+LowerDep** scheme significantly improved the performance of SpanBERT and ELECTRA on both datasets. ELECTRA-large with **Internal+LowerDep** achieved a new state-of-the-art performance on the OntoNotes and GAP benchmark.

For BERT-base, we suspect the topmost layer embeddings are no better than the penultimate layer embeddings because of the abnormal change of the three measures from the 11th layer to the 12th layer. Experiments show that the 12th layer embeddings of BERT-base are no better than the 11th layer embeddings for coreference resolution. The reasons are analyzed.

Experimental results also show that span embeddings from longer-context-encoded contextualized representations are better than higher-order span embeddings for coreference resolution.

References

[1] Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. SemEval-2012 task 6: A pilot on semantic textual similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393, Montréal, Canada, 7-8 June 2012. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/S12-1051.

[2] Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. *SEM 2013 shared task: Semantic textual similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 32–43, Atlanta, Georgia, USA, June 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/S13-1004.

[3] Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. SemEval-2014 task 10: Multilingual semantic textual similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 81–91, Dublin, Ireland, August 2014. Association for Computational Linguistics. doi: 10.3115/v1/S14-2010. URL https://www.aclweb.org/anthology/S14-2010.

[4] Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Iñigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, German Rigau, Larraitz Uria, and Janyce Wiebe. SemEval-2015 task 2: Semantic textual similarity, English, Spanish and pilot on interpretability. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 252–263, Denver, Colorado, June 2015. Association for Computational Linguistics. doi: 10.18653/v1/S15-2045. URL https://www.aclweb.org/anthology/S15-2045.

[5] Sanjeev Arora, Yingyu Liang, and Tengyu Ma. A simple but tough-to-beat baseline for sentence embeddings. In *Proceedings of International Conference on Learning Representations*, 2017.

[6] Amit Bagga and Breck Baldwin. Algorithms for scoring coreference chains. In *The first international conference on language resources and evaluation workshop on linguistics coreference*, volume 1, pages 563–566. Granada, 1998.

[7] Kevin Clark and Christopher D. Manning. Entity-centric coreference resolution with model stacking. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1405–1415, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1136. URL https://www.aclweb.org/anthology/P15-1136.

[8] Kevin Clark and Christopher D. Manning. Improving coreference resolution by learning entity-level distributed representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 643–653, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1061. URL https://www.aclweb.org/anthology/P16-1061.

[9] Kevin Clark and Christopher D. Manning. Deep reinforcement learning for mention-ranking coreference models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2256–2262, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1245. URL https://www.aclweb.org/anthology/D16-1245.

[10] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*, 2020. URL https://openreview.net/pdf?id=r1xMH1BtvB.

[11] Greg Durrett and Dan Klein. Easy victories and uphill battles in coreference resolution. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1971–1982, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D13-1203.

[12] Kawin Ethayarajh. How contextual are contextualized word representations? comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 55–65, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1006. URL https://www.aclweb.org/anthology/D19-1006.

[13] Hongliang Fei, Xu Li, Dingcheng Li, and Ping Li. End-to-end deep reinforcement learning based coreference resolution. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 660–665, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1064. URL https://www.aclweb.

org/anthology/P19-1064.

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[15] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. GloVe: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[16] Mandar Joshi, Omer Levy, Luke Zettlemoyer, and Daniel Weld. BERT for coreference resolution: Baselines and analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5803–5808, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1588. URL https://www.aclweb.org/anthology/D19-1588.

[17] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77, 2020.

[18] Ben Kantor and Amir Globerson. Coreference resolution with entity equalization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 673–677, Florence, Italy, July 2019. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P19-1066.

[19] Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1018. URL https://www.aclweb.org/anthology/D17-1018.

[20] Kenton Lee, Luheng He, and Luke Zettlemoyer. Higher-order coreference resolution with coarse-to-fine inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 687–692, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2108. URL https://www.aclweb.org/anthology/N18-2108.

[21] Xiaoqiang Luo. On coreference resolution performance metrics. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pages 25–32. Association for Computational Linguistics, 2005.

[22] Sebastian Martschat and Michael Strube. Latent structures for coreference resolution. *Transactions of the Association for Computational Linguistics*, 3:405–418, 2015. doi: 10.1162/tacl_a_00147. URL https://www.aclweb.org/anthology/Q15-1029.

[23] Jiaqi Mu, Suma Bhat, and Pramod Viswanath. All-but-the-top: Simple and effective postprocessing for word representations. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*, 2018.

[24] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1202. URL https://www.aclweb.org/anthology/N18-1202.

[25] Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. Conll-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes. In *Joint Conference on EMNLP and CoNLL-Shared Task*, pages 1–40. Association for Computational Linguistics, 2012.

[26] Karthik Raghunathan, Heeyoung Lee, Sudarshan Rangarajan, Nathanael Chambers, Mihai Surdeanu, Dan Jurafsky, and Christopher Manning. A multi-pass sieve for coreference resolution. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 492–501, Cambridge, MA, October 2010. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D10-1048.

[27] Cicero D Santos and Bianca Zadrozny. Learning character-level representations for part-of-speech tagging. In *Proceedings of the 31st international conference on machine learning (ICML-14)*, pages 1818–1826, 2014.

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[29] Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. A model-theoretic coreference scoring scheme. In *Proceedings of the 6th conference on Message understanding*, pages 45–52. Association for Computational Linguistics, 1995.

[30] Kellie Webster, Marta Recasens, Vera Axelrod, and Jason Baldridge. Mind the GAP: A balanced corpus of gendered ambiguous pronouns. *Transactions of the Association for Computational Linguistics*, 6:605–617, 2018. doi: 10.1162/tacl_a_00240. URL https://www.aclweb.org/anthology/Q18-1042.

[31] Sam Wiseman, Alexander M. Rush, Stuart Shieber, and Jason Weston. Learning anaphoricity and antecedent ranking features for coreference resolution. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1416–1426, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1137. URL https://www.aclweb.org/anthology/P15-1137.

[32] Sam Wiseman, Alexander M. Rush, and Stuart M. Shieber. Learning global features for coreference resolution. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 994–1004, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1114. URL https://www.aclweb.org/anthology/N16-1114.

[33] Rui Zhang, Cícero Nogueira dos Santos, Michihiro Yasunaga, Bing Xiang, and Dragomir Radev. Neural coreference resolution with deep biaffine attention by joint mention detection and mention clustering. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 102–107, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-2017. URL https://www.aclweb.org/anthology/P18-2017.

Chapter 6

Conclusion

This chapter provides some concluding remarks about this thesis. Throughout this thesis, we have made contributions to three subtasks of entity analysis: transfer learning for fine-grained entity typing (Chapter 2), entity linking using typed entity embeddings (Chapter 3), improving entity linking through anonymous entity mentions (Chapter 4), and exploiting less span embeddings for coreference resolution (Chapter 5). In this final chapter, we will recapitulate the proposed methods, and provide an outlook into the future.

6.1 Research Summary

In this dissertation, we have studied three sub-tasks of entity analysis: fine-grained entity typing, entity linking and entity coreference resolution. A recap of our methods and contributions is listed as follows.

Chapter 2 presented a new transfer learning based approach for fine-grained entity typing (FGET) that contains three transfer learning schemes. Firstly, to avoid on-site learning word vectors of out-of-vocabulary (OOV) words in mention phrases, we proposed to generate more precise word embeddings for OOVs through transfer learning using sub-word information. Secondly, instead of learning contextual features using LSTM, we proposed to generate contextual representations through transfer learning using a pre-trained bi-directional non-recurrent neural language model. Thirdly, to reduce the influence of label noises without twisting the original labels, we proposed to refine the predicted labels at inference time using a pre-trained topic model. The topic model associates types with topics through the so-called topic-anchors. The experimental results on two standard FGET corpora validated the effectiveness of our

transfer learning approach. Compared with previous methods, our method can predict more fine-grained labels and achieve the state-of-the-art performance.

Chapter 3 presented a simple yet effective method, FGS2EE, to inject fine-grained semantic information into entity embeddings to reduce the distinctiveness and facilitate the learning of contextual commonality. FGS2EE first uses the word embeddings of semantic type words to generate semantic embeddings, and then combines them with existing entity embeddings through linear aggregation. Our entity embeddings draw entities of similar types closer, while entities of different types are drawn further. Thus can facilitate the learning of semantic commonalities about entity-context and entity-entity relations. We have achieved new state-of-the-art performance using our entity embeddings.

Chapter 4 presented a method DOC-AET to exploit DOCument-level coherence of named entity mentions and anonymous entity type (AET) words/mentions for improving named entity linking. We show that incorporating the coherence score between candidate entities and AET mentions can significantly improve NEL performance. DOC-AET used the fine-grained type words of Hou et al. [5] as AET vocabulary to extract anonymous entity mentions. The document-level relatedness between entity types is encoded into the AET word embeddings which are learnt from the AET words' inter-paragraph co-occurrence matrix. AET entity embeddings and document AET context embeddings are computed using the AET word embeddings. The coherence scores between candidate entities and anonymous entities are computed using the AET entity embeddings and document context embeddings. By incorporating such coherence scores for candidate ranking, we achieve state-of-the-art performance on three of the five out-domain datasets.

Chapter 5 proposed five LASE (Less Anisotropic Span Embeddings) schemes to generate less *anisotropic* span embeddings for coreference resolution. Before applying LASE schemes, we use three measures to gauge the contextualized representations of BERT [6], SpanBERT [7] and ELECTRA [1]. The **Internal+LowerDep** scheme significantly improved the performance of SpanBERT and ELECTRA on both datasets. ELECTRA-large with **Internal+LowerDep** achieved a new state-of-the-art performance on the OntoNotes and GAP benchmark. For BERT-base, we suspect the topmost layer embeddings are no better than the penultimate layer embeddings because of the abnormal change of the three measures from the 11th layer to the 12th layer. Experiments show that the 12th layer embeddings of BERT-base are no better

than the 11th layer embeddings for coreference resolution. The reasons are analyzed. Experimental results also show that span embeddings from longer-context-encoded contextualized representations are better than higher-order span embeddings for coreference resolution.

6.2 Future Directions

In this section, we will give an outlook into the future for research on entity analysis.

Multi-task learning for entity linking and cluster-ranking coreference resolution. Multi-task learning, motivated by Stein's paradox [10], is a learning paradigm in which data from multiple tasks is used with the hope to obtain superior performance over learning each task independently. Multi-task learning in natural language processing [2], [4], is typically conducted via hard parameter sharing among sequence labelling tasks. In hard parameter sharing, a subset of the parameters is shared between tasks while other parameters are task-specific. Typically, these parameters are learned by solving an optimization problem that minimizes a weighted sum of the empirical risk for each task. Sener and Koltun [9] formulate multi-task learning as multi-objective optimization (MOO) problem and they apply gradient-based MOO to multi-task learning.

Entity linking and coreference resolution are synergistic as demonstrated by Durrett and Klein [3]. On the one hand, coreferent clusters can provide more information for entity linking. On the other hand, coreference resolution can also benefit from incremented knowledge through linked entity mentions. For example, knowing from knowledge base (KB) that *Donald Trump* is a *U.S. president* would be helpful for establishing the coreference relation between the two mentions. However, it is difficult to obtain such information from KB without performing entity linking.

A multi-objective reinforcement learning (MORL) based joint model for entity linking and coreference resolution. Unlike Durrett and Klein [3], who use factor graph as a joint model for entity analysis, we can put entity linking and coreference resolution under the multi-objective reinforcement learning paradigm [8]. This joint model makes a sequence of decisions about entity linking and coreference resolution. Firstly mentions that are easy to resolve are resolved. Then entity linking is performed using coreferent information. Repeat resolving the mentions that are difficult to resolve using KB information of linked entities. Both tasks will benefit from incremented information.

References

[1] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*, 2020. URL https://openreview.net/pdf?id=r1xMH1BtvB.

[2] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM, 2008.

[3] Greg Durrett and Dan Klein. A joint model for entity analysis: Coreference, typing, and linking. *Transactions of the association for computational linguistics*, 2:477–490, 2014.

[4] Kazuma Hashimoto, Caiming Xiong, Yoshimasa Tsuruoka, and Richard Socher. A joint many-task model: Growing a neural network for multiple NLP tasks. pages 1923–1933, September 2017. doi: 10.18653/v1/D17-1206. URL https://www.aclweb.org/anthology/D17-1206.

[5] Feng Hou, Ruili Wang, Jun He, and Yi Zhou. Improving entity linking through semantic reinforced entity embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6843–6848, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.612. URL https://www.aclweb.org/anthology/2020.acl-main.612.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[7] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77, 2020.

[8] Chunming Liu, Xin Xu, and Dewen Hu. Multiobjective reinforcement learning: A comprehensive overview. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45 (3):385–398, 2015.

[9] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. In *Advances in Neural Information Processing Systems*, pages 527–538, 2018.

[10] Charles Stein. Inadmissibility of the usual estimator for the mean of a multivariate normal distribution. In *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability*, page 197–206, 1956.

Appendix A

Statement of Contribution

I confirm that the "Statement of Contribution to Doctoral Thesis Containing Publications (DRC16)", have been completed for each published article within the thesis, and are bound into the thesis and included in the electronic copy.

MASSEY UNIVERSITY
TE KUNENGA KI PŪREHUROA
UNIVERSITY OF NEW ZEALAND

GRADUATE RESEARCH SCHOOL

STATEMENT OF CONTRIBUTION
DOCTORATE WITH PUBLICATIONS/MANUSCRIPTS

We, the candidate and the candidate's Primary Supervisor, certify that all co-authors have consented to their work being included in the thesis and they have accepted the candidate's contribution as indicated below in the *Statement of Originality*.

Name of candidate:	Feng Hou
Name/title of Primary Supervisor:	Professor Ruili Wang
In which chapter is the manuscript /published work:	Chapter 2

Please select one of the following three options:

○ The manuscript/published work is published or in press

- Please provide the full reference of the Research Output:

● The manuscript is currently under review for publication – please indicate:

- The name of the journal:

Knowledge and Information Systems

- The percentage of the manuscript/published work that was contributed by the candidate: 75.00

- Describe the contribution that the candidate has made to the manuscript/published work:

- Proposed three transfer learning schemes for fine-grained entity typing
- Implemented the experiments on two data sets

○ It is intended that the manuscript will be published, but it has not yet been submitted to a journal

Candidate's Signature:	Feng Hou 数字签名者：Feng Hou DN：cn=Feng Hou, o=Massey University, ou=School of Natural and Computational Sciences, email=f.hou@massey.ac.nz, c=NZ 日期：2020.10.29 11:48:54 +13'00'
Date:	29/10/2020
Primary Supervisor's Signature:	Prof Ruili Wang Digitally signed by Prof Ruili Wang DN: cn=Prof Ruili Wang, c=NZ, o=Massey University , ou=School of Natural and Computational Sciences , email=ruili.wang@massey.ac.nz Date: 2020.11.04 17:07:14 +13'00'
Date:	

This form should appear at the end of each thesis chapter/section/appendix submitted as a manuscript/ publication or collected as an appendix at the end of the thesis.

MASSEY UNIVERSITY
TE KUNENGA KI PŪREHUROA
UNIVERSITY OF NEW ZEALAND

GRADUATE RESEARCH SCHOOL

STATEMENT OF CONTRIBUTION
DOCTORATE WITH PUBLICATIONS/MANUSCRIPTS

We, the candidate and the candidate's Primary Supervisor, certify that all co-authors have consented to their work being included in the thesis and they have accepted the candidate's contribution as indicated below in the *Statement of Originality*.

Name of candidate:	Feng Hou
Name/title of Primary Supervisor:	Professor Ruili Wang
In which chapter is the manuscript /published work:	Chapter 3

Please select one of the following three options:

(●) The manuscript/published work is published or in press

- Please provide the full reference of the Research Output:

Feng Hou, Ruili Wang, Jun He and Yi Zhou. 2020. Improving Entity Linking through Semantic Reinforced Entity Embeddings. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 6843– 6848, July 5 - 10, 2020. Association for Computational Linguistics.

( ) The manuscript is currently under review for publication – please indicate:

- The name of the journal:

- The percentage of the manuscript/published work that was contributed by the candidate:

- Describe the contribution that the candidate has made to the manuscript/published work:

( ) It is intended that the manuscript will be published, but it has not yet been submitted to a journal

Candidate's Signature:	Feng Hou 数字签名者：Feng Hou DN：cn=Feng Hou, o=Massey University, ou=School of Natural and Computational Sciences, email=f.hou@massey.ac.nz, c=NZ 日期：2020.10.29 11:52:09 +13'00'
Date:	29/10/2020
Primary Supervisor's Signature:	Prof Ruili Wang Digitally signed by Prof Ruili Wang DN: cn=Prof Ruili Wang, c=NZ, o=Massey University , ou=School of Natural and Computational Sciences , email=ruili.wang@massey.ac.nz Date: 2020.11.04 17:07:51 +13'00'
Date:	

This form should appear at the end of each thesis chapter/section/appendix submitted as a manuscript/ publication or collected as an appendix at the end of the thesis.

MASSEY UNIVERSITY
TE KUNENGA KI PŪREHUROA
UNIVERSITY OF NEW ZEALAND

GRADUATE RESEARCH SCHOOL

STATEMENT OF CONTRIBUTION
DOCTORATE WITH PUBLICATIONS/MANUSCRIPTS

We, the candidate and the candidate's Primary Supervisor, certify that all co-authors have consented to their work being included in the thesis and they have accepted the candidate's contribution as indicated below in the *Statement of Originality*.

Name of candidate:	Feng Hou
Name/title of Primary Supervisor:	Professor Ruili Wang
In which chapter is the manuscript /published work:	Chapter 4

Please select one of the following three options:

○ The manuscript/published work is published or in press

- Please provide the full reference of the Research Output:

● The manuscript is currently under review for publication – please indicate:

- The name of the journal:

35th AAAI Conference on Artificial Intelligence

- The percentage of the manuscript/published work that was contributed by the candidate: 75.00

- Describe the contribution that the candidate has made to the manuscript/published work:

- Proposed an idea of using the coherence between named entities and anonymous entity mentions
- Implemented the method of incorporating the proposed coherence
- Implemented the experiments on two data sets

○ It is intended that the manuscript will be published, but it has not yet been submitted to a journal

Candidate's Signature:	Feng Hou 数字签名者：Feng Hou DN：cn=Feng Hou, o=Massey University, ou=School of Natural and Computational Sciences, email=f.hou@massey.ac.nz, c=NZ 日期：2020.10.29 11:57:15 +13'00'
Date:	29/10/2020
Primary Supervisor's Signature:	Prof Ruili Wang Digitally signed by Prof Ruili Wang DN: cn=Prof Ruili Wang, c=NZ, o=Massey University , ou=School of Natural and Computational Sciences , email=ruili.wang@massey.ac.nz Date: 2020.11.04 17:06:41 +13'00'
Date:	

This form should appear at the end of each thesis chapter/section/appendix submitted as a manuscript/ publication or collected as an appendix at the end of the thesis.

MASSEY UNIVERSITY
TE KUNENGA KI PŪREHUROA
UNIVERSITY OF NEW ZEALAND

GRADUATE RESEARCH SCHOOL

STATEMENT OF CONTRIBUTION
DOCTORATE WITH PUBLICATIONS/MANUSCRIPTS

We, the candidate and the candidate's Primary Supervisor, certify that all co-authors have consented to their work being included in the thesis and they have accepted the candidate's contribution as indicated below in the *Statement of Originality*.

Name of candidate:	Feng Hou
Name/title of Primary Supervisor:	Professor Ruili Wang
In which chapter is the manuscript /published work:	Chapter 5

Please select one of the following three options:

☐ The manuscript/published work is published or in press

- Please provide the full reference of the Research Output:

☐ The manuscript is currently under review for publication – please indicate:

- The name of the journal:

- The percentage of the manuscript/published work that was contributed by the candidate:

- Describe the contribution that the candidate has made to the manuscript/published work:

☉ It is intended that the manuscript will be published, but it has not yet been submitted to a journal

Candidate's Signature:	Feng Hou 数字签名者：Feng Hou DN：cn=Feng Hou, o=Massey University, ou=School of Natural and Computational Sciences, email=f.hou@massey.ac.nz, c=NZ 日期：2020.10.29 11:58:18 +13'00'
Date:	29/10/2020
Primary Supervisor's Signature:	Prof Ruili Wang Digitally signed by Prof Ruili Wang DN: cn=Prof Ruili Wang, c=NZ, o=Massey University , ou=School of Natural and Computational Sciences , email=ruili.wang@massey.ac.nz Date: 2020.11.04 17:08:27 +13'00'
Date:	

This form should appear at the end of each thesis chapter/section/appendix submitted as a manuscript/ publication or collected as an appendix at the end of the thesis.