

Université de Montréal

Leveraging Distant Supervision for Improved Named Entity Recognition

par
Abbas Ghaddar

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Thèse présentée à la Faculté des études supérieures
en vue de l'obtention du grade de Philosophiæ Doctor (Ph.D.)
en computer science

Mars, 2020

© Abbas Ghaddar, 2020.

RÉSUMÉ

Les techniques d'apprentissage profond ont fait un bond au cours des dernières années, et ont considérablement changé la manière dont les tâches de traitement automatique du langage naturel (TALN) sont traitées. En quelques années, les réseaux de neurones et les plongements de mots sont rapidement devenus des composants centraux à adopter dans le domaine.

La supervision distante (SD) est une technique connue en TALN qui consiste à générer automatiquement des données étiquetées à partir d'exemples partiellement annotés. Traditionnellement, ces données sont utilisées pour l'entraînement en l'absence d'annotations manuelles, ou comme données supplémentaires pour améliorer les performances de généralisation.

Dans cette thèse, nous étudions comment la supervision distante peut être utilisée dans un cadre d'un TALN moderne basé sur l'apprentissage profond. Puisque les algorithmes d'apprentissage profond s'améliorent lorsqu'une quantité massive de données est fournie (en particulier pour l'apprentissage des représentations), nous revisitons la génération automatique des données avec la supervision distante à partir de Wikipédia. On applique des post-traitements sur Wikipédia pour augmenter la quantité d'exemples annotés, tout en introduisant une quantité raisonnable de bruit.

Ensuite, nous explorons différentes méthodes d'utilisation de données obtenues par supervision distante pour l'apprentissage des représentations, principalement pour apprendre des représentations de mots classiques (statistiques) et contextuelles.

À cause de sa position centrale pour de nombreuses applications du TALN, nous choisissons la reconnaissance d'entité nommée (NER) comme tâche principale. Nous expérimentons avec des bancs d'essai NER standards et nous observons des performances état de l'art. Ce faisant, nous étudions un cadre plus intéressant, à savoir l'amélioration des performances inter-domaines (généralisation).

Mots clés: Supervision distante, Wikipédia, Représentation de mots, NER, Généralisation.

ABSTRACT

Recent years have seen a leap in deep learning techniques that greatly changed the way Natural Language Processing (NLP) tasks are tackled. In a couple of years, neural networks and word embeddings quickly became central components to be adopted in the domain.

Distant supervision (DS) is a well used technique in NLP to produce labeled data from partially annotated examples. Traditionally, it was mainly used as training data in the absence of manual annotations, or as additional training data to improve generalization performances.

In this thesis, we study how distant supervision can be employed within a modern deep learning based NLP framework. As deep learning algorithms gets better when massive amount of data is provided (especially for representation learning), we revisit the task of generating distant supervision data from Wikipedia. We apply post-processing treatments on the original dump to further increase the quantity of labeled examples, while introducing a reasonable amount of noise.

Then, we explore different methods for using distant supervision data for representation learning, mainly to learn classic and contextualized word representations. Due to its importance as a basic component in many NLP applications, we choose Named-Entity Recognition (NER) as our main task. We experiment on standard NER benchmarks showing state-of-the-art performances. By doing so, we investigate a more interesting setting, that is, improving the cross-domain (generalization) performances.

Keywords: Distant Supervision, Wikipedia, Word Representation, NER, Generalization

CONTENTS

RÉSUMÉ	ii
ABSTRACT	iii
CONTENTS	iv
LIST OF TABLES	viii
LIST OF FIGURES	xii
ACKNOWLEDGMENTS	xvii
CHAPTER 1: INTRODUCTION	1
1.1 Problem Statement	1
1.2 Contributions	5
1.3 Previously Published Material	6
1.4 Thesis Structure	7
CHAPTER 2: REPRESENTATION LEARNING	9
2.1 Word embeddings	9
2.2 Contextualized Word Representation	14
2.3 Fine tuning Approaches	18
2.4 Conclusion	22
CHAPTER 3: DISTANT SUPERVISION FOR NER	23
3.1 Named Entity Recognition	23
3.2 Datasets	24
3.3 Approaches to NER	26
3.4 Metrics	28
3.5 Distant Supervision for NER	29

3.5.1	Web page based Corpora	30
3.5.2	Wikipedia for NER	31
3.6	Enriching Wikipedia with Links	34
3.6.1	Recent works on DS	35
3.7	Conclusion	36
CHAPTER 4: FROM WIKIPEDIA TO WINER AND WIFINE		38
4.1	Overview	38
4.2	A Four Stage Approach	39
4.2.1	Main Concept Mention Detection	40
4.2.2	Secondary Entities Mentions Detection	41
4.2.3	Manual evaluation of link augmentation	43
4.3	WiNER	44
4.4	WiFiNE	45
4.4.1	Corpus Statistics	51
4.5	Conclusion	53
CHAPTER 5: DISTANT SUPERVISION DATA FOR TRAINING		54
5.1	Overview	54
5.2	Experiments on WiNER	54
5.2.1	Data Sets	55
5.2.2	Reference Systems	56
5.2.3	Metrics	56
5.2.4	Comparing with other Wikipedia-based Corpora	57
5.2.5	Cross-domain Evaluation	58
5.2.6	Scaling up to WiNER	60
5.3	Experiments on WiFiNE	65
5.3.1	Reference System	65
5.3.2	Datasets and Evaluation Metrics	66
5.3.3	Results on FIGER (GOLD)	67
5.3.4	Results on OntoNotes	68

5.4	Conclusion	71
CHAPTER 6: ROBUST LEXICAL FEATURES FOR IMPROVED NEURAL NETWORK NAMED-ENTITY RECOGNITION . . . 72		
6.1	Overview	72
6.2	Introduction	73
6.3	Motivation	74
6.4	Our Method	75
6.4.1	Corpus Description	75
6.4.2	Embedding Words and Entity Types	75
6.4.3	LS Representation	77
6.4.4	Strength of the LS Representation	79
6.5	Our NER System	79
6.5.1	Bi-LSTM-CRF Model	79
6.5.2	Features	79
6.6	Experiments	82
6.6.1	Data and Evaluation	82
6.6.2	Training and Implementation	82
6.6.3	Results on Dev	83
6.6.4	Results on CONLL	84
6.6.5	Results on ONTONOTES	85
6.6.6	Ablation Results	87
6.7	Related Works	88
6.8	Conclusion	88
CHAPTER 7: CONTEXTUALIZED WORD REPRESENTATIONS FROM DISTANT SUPERVISION WITH AND FOR NER 90		
7.1	Overview	90
7.2	Introduction	90
7.3	Data and Preprocessing	91
7.4	Learning our Representation	92

7.5	Experiments on NER	95
7.5.1	Datasets	95
7.5.2	Input Representations	95
7.6	Experiments	97
7.6.1	Comparison to LS embeddings	97
7.6.2	Comparing Contextualized Embeddings	98
7.6.3	Analysis	100
7.7	Conclusion	101
CHAPTER 8: CONCLUSION AND FUTURE WORKS		102
8.1	Future Work	103
BIBLIOGRAPHY		108

LIST OF TABLES

1.I	Accuracy (F1 score) from the experiments done by Alvarado et al. [6] on the impact of domain mismatch on Named entity Recognition models performances. Figures between parentheses show the size of the datasets in term of number of tokens.	2
2.I	Gains over state-of-the-art models (at that time) by adding ELMo [120] as feature to single model baselines across six benchmark NLP tasks: Reading Comprehension (SQuAD), Textual Entailment(SNLI), Semantic Role Labeling (SRL), Coreference Resolution (Coref), Named Entity Recognition (NER), and Sentiment Analysis (SST-5).Source [120]	16
3.I	List of manually annotated datasets for English NER, with the number of entity types (#Tags). source [86].	25
3.II	Entity level F1 scores on test sets of CONLL-2003 and ONTONOTES 5.0 respectively. The model description show the main contribution of the paper. The first set are feature-based (classic) models; the second set are neural models without external knowledge as feature; the last set are neural models with external knowledge. . .	29
3.III	Comparison of different link-enriched corpora. Counts (# columns) are in millions.	35
4.I	Comparison of different link-enriched corpora. Counts in columns Links, Entities and Documents are in millions.	43
4.II	Number of times a text string (mention) is labelled with (at least) two types in WiNER. The cells on the diagonal indicate the number of annotations.	45
4.III	De-noising rules evaluation on 1000 hand-labelled mentions following GILLICK type hierarchy.	49

4.IV	Random selection of annotations from WiFiNE following GILLICK type hierarchy. Faulty annotations are marked with a star.	50
4.V	Mention statistics and label distribution (in millions and percentages) over the number of levels of FIGER and GILLICK type hierarchy.	52
5.I	Performance of the <code>Illinois</code> toolkit on CONLL, as a function of the Wikipedia-based training material used. The figures on the last line are averaged over the 10 subsets of WiNER we randomly sampled. Bracketed figures indicate the minimum and maximum values.	57
5.II	Cross-domain evaluation of NER systems trained on different mixes of CONLL and WiNER. Figures are token-level F1 score on 3 classes, while figures in parentheses indicate absolute gains over the configuration using only the CONLL training material.	58
5.III	Features for the segment <i>Gonzales</i> in the sentence <i>Gonzales will be featured on Daft Punk</i>	61
5.IV	Influence of the portion of WiNER used in our 2-stage approach for the CONLL test set, using the segmentation produced by <code>LSTM-CRF+WiNER(5M)</code> . These results have to be contrasted with the last line of Table 5.II.	62
5.V	OD_{F1} score of native configurations, and of our 2-stage approach (RF) which exploits the full WiNER corpus. Figures in parentheses indicate absolute gains over the native configuration.	63
5.VI	Percentage of correctness of the 2-stage system (rows) when tagging a named-entity differently than the <code>LSTM-CRF+WiNER(5M)</code> (columns). Bracketed figures indicate the average number of differences over the out-domain test sets.	64

5.VII	Results of the reference system trained on various subsets of WiFiNE, compared to other published results on the FIGER (GOLD) test set. Training data (in millions) include: proper name; nominal and pronominal mentions.	67
5.VIII	Comparison of the distribution of the top 5 types present in FIGER (GOLD) test set to that of WiFiNE.	68
5.IX	Results of the reference system trained on various subsets of WiFiNE, compared to other published results on the ONTONOTES test set. Training data (in millions) includes: proper name; nominal and pronominal mentions.	69
5.X	Comparison of the distribution of the top 5 types present in ONTONOTES test set to that of WiFiNE.	70
5.XI	Examples of non-entity mentions annotated as /other in the of OntoNotes test set.	70
6.I	Topmost similar entity types to a few single-word mentions (first four) and non-entity words (last four).	78
6.II	Statistics of the CONLL-2003 and ONTONOTES 5.0 datasets. #tok stands for the number of tokens, and #ent indicates the number of named-entities gold annotated.	83
6.III	Development set F1 scores of our best hyper-parameter setting compared to the results reported by [30].	84
6.IV	F1 scores on the CONLL test set. The first four systems are feature-based, the others are neuronal. The LGCEcMS column indicates the feature configuration of each system. L stands for Lexical feature, G for Gazetteers, C for Capitalization, E for pre-trained Embeddings, c for character embeddings, M for language Model Embeddings, and S for the proposed LS feature representation. + indicates that the model use the feature set.	85

6.V	Per-genre F1 scores on ONTONOTES (numbers taken from Chiu and Nichols [30]). BC = broadcast conversation, BN = broadcast news, MZ = magazine, NW = newswire, TC = telephone conversation, WB = blogs and newsgroups.	85
6.VI	F1 scores on the ONTONOTES test set. The first four systems are feature-based, the following ones are neuronal. See Table 6.IV for an explanation of the LGCEcMS column.	86
6.VII	F1 scores of differently trained systems on CONLL-2003 and ONTONOTES 5.0 datasets. Capitalization (Section 6.5.2.3) and character features (Section 6.5.2.2) are used by default by all models.	87
7.I	Statistics on the datasets used in our experiments.	95
7.II	F1 scores over five runs on CONLL and ONTONOTES test set of ablation experiments. We evaluate 4 baselines without additional embeddings (column \mathcal{X}) and with LS embeddings [57] or ours. Figures in parenthesis indicate the gain over the baselines.	98
7.III	Mention-level F1 scores. The baseline (first line) uses word shape and traditional (classic) embeddings. Variants stacking various representations are presented in decreasing order of F1 return. So for instance, ELMo is the best representation to add to the baseline one.	99

LIST OF FIGURES

2.1	Illustration of the CBOW and the Skip-gram models proposed by Mikolov et al. [104]. Source [104]	10
2.2	Two-dimensional representation of the vector space of <i>word2vec</i> embeddings of selected cities and their respective capitals. Source [104]	11
2.3	word-word co-occurrence probabilities and ratio between the target words (<i>ice</i> and <i>steam</i>) and 4 selected context words. Source [118].	12
2.4	Example of a sentence with dependency labels used in the work of Levy and Goldberg [85]. Source [85]	13
2.5	Sentences with dependency labels from Neelakantan et al. [108], where parse tree is used to obtain arbitrary context words. Source [108]	13
2.6	The general framework of language model pre-training. Given a sentence, the goal is to predict, at each step, the next word given the previous context. A sequence encoder such as LSTM or Transformer is employed to encode syntactic and semantic features. The hidden state are considered as <i>contextualized word representations</i> because they varies for each word depending on the context. . . .	14
2.7	Context-based representations generated by a neural language model for an input sentence of a specific supervised task (NER, QA,...). .	16
2.8	Left Figure: An encoder decoder neural machine translation model is trained offline. Right Figure: the encoder is used to obtain representations, which in turn are used as input for supervised tasks. Source [99]	17
2.9	Key differences between BERT [41], OpenAI GPT [127] and Elmo[120] pre-training objectives. Source [41]	18

2.10	Illustration of an input sequence of BERT [41]. Some words are randomly masked in the input sentence and the goal is to predict them at the output. No masked word are ignored in the loss calculation (transparent rectangle).	19
2.11	Illustration of how BERT[41] can be fine tuned for a wide range of NLP tasks. Source [41]	20
3.1	An illustration of the named entity recognition task. Given a sentence, the goal is to identify entity mentions and to classify them into predefined categories. The top part shows NER with coarse categories, while the bottom part shows the task with more fine grained types.	23
3.2	High level architecture for NER model as a sequence labeling problem. Each word in the input is represented by a set of features. These features are fed into a classifier (gray box) which in turn produces a label per token at the output layer indicating the entity type of the token.	26
3.3	Excerpt from the Wikipedia article <i>Barack Obama</i>	31
3.4	Excerpt of the Freebase page of <i>Barack Obama</i>	32
3.5	Wikipedia to named-entity annotated corpus pipeline of Nothman et al. [112]. source [112]	33
4.1	Illustration of the process with which we gather annotations into WiNER for the target page https://en.wikipedia.org/wiki/Chilly_Gonzales . Square Bracketed segments are the annotations; curly brackets indicate main concept mentions from Ghaddar and Langlais [55]; while underlined text are anchored texts in the corresponding Wikipedia page. OLT represents the out-link table (which is compiled from the Wikipedia out-link graph structure), and CT represents the coreference table we gathered from the resource.	39

4.2	The first step of our process: main concept mentions detection. Given the article of Chilly Gonzales the goal is to find all nominal and pronominal mentions (red span) that refer to <i>Chilly Gonzales</i> .	40
4.3	Two examples that illustrate the first step of our process: link detection by following link out of link out.	42
4.4	The second step of our process: proper and nominal coreference mention detection. The text box is our outgoing example, and the bullet list contains coreference mention of entities that we match in the paragraph (e.g. <i>Warner Bros.</i>).	42
4.5	All entity links in our outgoing example are mapped to 4 entity types according to CONLL-2003 annotation scheme.	44
4.6	Illustration of mapping entity link (Paris in this example) to named entity annotation through Freebase.	45
4.7	Illustration of mapping entity link (Paris in this example) to hierarchical fine grained entity types through Freebase.	46
4.8	(a) FIGER [91] annotation scheme consists of 112 entity types that are stored in a 2 level hierarchical structure (red rectangles indicate parent types). (b) GILLICK [58] defines 3 levels of types, a total of 89 labels (separated by boxes). source [91] and [58]	47
4.9	Illustration of the de-noise heuristics rules. Spans in bold are entity mentions, and blue labels are relevant labels, while red are irrelevant ones.	48
4.10	Examples of errors in our de-noising rules. Faulty annotations are marked with a star.	51
4.11	Distribution of entity type labels according to the FIGER type hierarchy.	52
5.1	Two representations of WiNER's annotation used for feature extraction.	60
5.2	Example of entities re-classified by our 2-stage approach.	64

5.3	An illustration of the attentive encoder neural model of Shimaoka et al. [143] predicting fine-grained semantic types for the mention “New Zealand” in the expression “a match series against New Zealand is held on Monday”. source [143]	65
5.4	Examples of mentions erroneously classified in FIGER (GOLD) dataset.	68
6.1	An example from Chiu and Nichols [30] that show the limitation of binary encoded gazetteer features. For instance, the word <i>China</i> appear as entry in 4 lists of named entities, in real world it mostly refer to the country. However, binary features evenly attribute the same weight for the 4 classes. source [30]	74
6.2	Example of the two variants of a given sentence.	76
6.3	Two-dimensional representation of the vector space which embeds both words and entity types. Big Xs indicate entity types, while circles refer to words (i.e. named entities, here).	77
6.4	Main architecture of our NER system.	80
6.5	Character representation of the word « Roma » given to the bidirectional LSTM of Figure 6.4.	81
7.1	Input (rose) and output (yellow) sequences used by our encoder to learn contextualized representations. Transparent yellow box indicates that no prediction is made for the corresponding token. .	91
7.2	Illustration of the architecture of the model used for learning our representation. It consists of stacked layers of dilated convolutional neural network followed by a self-attention layer. The input is a sequence of tokens with a maximum length of 512, where the output is the associated entity type sequence. We use the hidden state of the last DCNN layer and the self-attention layer as our representation.	92

7.3	Difference between Regular (sub-figure a) and Dilated (sub-figure b) CNN. Like typical CNN layers, dilated convolutions operate on a sliding window of context over the sequence, but unlike conventional convolutions, the context need not be consecutive.	93
7.4	An artificial example that illustrate how attention weights of a self-attention layer can behave if the outputs to predict are entity types. For example, the word <i>born</i> that precedes <i>John</i> is a strong indicator that the latter should be tagged as person. Similarly, for the target word <i>Montreal</i> and its context <i>born in</i>	94
7.5	Main architecture of NER model used in this work. Green circle at the input layer show baseline features, while blue ones show contextualized word representations that are added.	96
7.6	An illustration on how we analyze the impact of self-attention in influencing document context. We check the words that received the highest attention weights inside each document for CONLL dev portion.	100
7.7	top 5 attended words for some randomly picked documents in the dev set of CONLL. Column 1 indicate document number, while column 2 is our appreciation of the document topic.	100
8.1	Example of correct (a-b) and wrong (c-d) relations annotated using distant supervision.	106

ACKNOWLEDGMENTS

I am deeply grateful to Professor Philippe Langlais who is a fantastic supervisor. The last 6 years has been intellectually stimulating, rewarding and fun. He has gently shepherded my research down interesting paths. I hope that I have managed to absorb just some of his dedication and taste for research. It has been a true privilege.

I have been very lucky to meet and interact with the extraordinarily skillful Fabrizio Gotti who kindly helped me to debug code when I got stuck on some computer problems.

Many thanks also to the members of RALI-lab, a group of friends and colleagues, that I have been fortunate enough to be surrounded with. Finally, I would like to thank my dearest parents, aunt and uncle for being unwavering in their support.

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp and Titan V GPUs used for this research.

CHAPTER 1

INTRODUCTION

1.1 Problem Statement

Deep Learning algorithms [81] appeared to be a very powerful techniques, leading to impressive advances in various fields, such as computer vision, and speech processing. Natural Language Processing (NLP) is not an exception, indeed, neural networks architectures (such as Recurrent [102] and Convolutional [74] Neural Networks, and Transformer [160]) became a natural choice to model a large scale of tasks. Since its emergence in 2010, transfer learning powered by word representation learning techniques such `word2vec` [104] in 2013; ELMo [120] in 2017; and BERT [41] in 2019 have and continue to prove that it is a practical, powerful and task-independent solution to boost performances with minimal human intervention. The approach consists of training an encoder on large scale unlabeled data and transferring the learned representations to another supervised task (trained on labeled data).

Deep learning algorithms are data hungry [12, 105], that is, good performances are observed when large scale, manually annotated, in domain data is available. Thus, the advancement in deep learning algorithms for NLP was accompanied by an increasing interest in creating large scale crowd sourced¹ gold benchmarks that support multiple tasks such as: SST [146] for sentiment analysis; SQUAD1.0 [130], SQUAD2.0 [131] and RACE [77] for question answering; SNLI [22] and MNLI [166] for natural language inference.

In real world applications, machine learning models must be able to generalize from small amount of in-domain or even out-of-domain data in some case, mainly because human annotation is an expensive, time consuming and error prone process. Despite the advancement made in representation learning and deep algorithms, multiple studies have shown that current models are still data dependent.

1. mostly using Amazon mechanical Turk services

Train/Test	CoNLL	Finance
CoNLL (240 tokens)	83	17
Finance (40 tokens)	-	83
CoNLL + Finance	-	79

Table 1.I – Accuracy (F1 score) from the experiments done by Alvarado et al. [6] on the impact of domain mismatch on Named entity Recognition models performances. Figures between parentheses show the size of the datasets in term of number of tokens.

Table 1.I, shows the results of domain mismatch experiments done by Alvarado et al. [6] on Named Entity Recognition (see Chapter 3 for an overview on the task). The authors compared the performances of models trained and tested on news (the CoNLL dataset [154]) and finance data. The authors observe that the model trained on news perform poorly (17%) when tested on finance compared to the one trained on finance (83%). Furthermore, the system trained on finance only (83%) outperforms the one trained on both news and finance (79%) when tested on finance (despite being trained on 6 times more data).

An extensive empirical study by Augenstein et al. [10] on memorization and generalization of current NER systems confirm the previous observation. Also, a recent study by McCoy et al. [100] has shown that the state-of-the-art representation learning model BERT [41] rely on heuristics that are effective for frequent examples in SNLI benchmark [22] to solve textual entailment (a binary classification task). Consequently, the model breaks down on a more challenging dataset where heuristics fail. In nutshell, the out-of-domain (generalization) performance remains an old-new problem in NLP.

The small amount and limited scope of annotated data available for training NLP systems, have motivated some researchers to create a large-scale automatically labeled corpus. Distant supervision techniques are very promising as they can be used to overcome the lack of large-scale labelled data in NLP applications. This technique mainly consists in generating training data out of partly annotated examples (e.g. Wikipedia) that are linked to a knowledge base (such as Freebase). Surface forms of links that point to knowledge base entries are considered the main source of annotations, which in turn are used to train supervised models as by Mintz et al. [106] for Relation Extrac-

tion (RE), Al-Rfou et al. [5], Nothman [111] for NER and [91] for Fine-Grained Entity Typing (FGET).

In this thesis, we revisit distant supervision in the light of recent advancements in deep learning techniques for NLP. Otherwise said, we study how distant supervision can be employed within a modern deep learning based NLP framework. When we analyzed this subject thoroughly, we identified two possibilities for contribution:

- Revisit the automatic annotation process in order to get more data.
- Study the utility of distant supervision data for word representation learning.

Wikipedia is widely used as the backbone for distant supervision, mainly due to its availability, structure, diversity and the presence of human annotations (e.g. hyperlinks and infoboxes). However, most prior works use the resource as is (direct map of hyperlinks), or increment the number of hyperlinks with simple heuristics (see Chapter 3).

In our opinion, this is mainly due to the fact that data played a much less important role at that time. In the last ten years, distant supervision was intended to be used to train classic machine learning systems that require a limited amount of data and relies more on feature engineering. However, the emergence of deep learning algorithms and the increasing availability of computational resources have changed the rules. Features are learned directly from data, and the more data we have, the better deep learning models can perform.

Motivated by the increasing need of large scale annotated data, it was important to revisit distant supervision techniques applied to Wikipedia. We argue that the resource has a special structure and characteristics that allow to mine much more annotations than by only relying on human made hyperlinks. We propose heuristics rules that are adapted to the nature of Wikipedia in order to enrich it with hyperlinks. Our main objective is to gather as many annotations as possible, while introducing the less amount of noise.

At the time of writing this thesis, two very recent papers have been introducing an alternative approach to improve distant supervision for NER [178] and FGET [3] using deep models as annotator (more details Chapter 3). They do not compare directly with our methods, which we leave for future comparisons. However, these works reveal the increasing interest in distant supervision as a reasonable solution to acquire cheap yet

useful annotated data.

Traditionally, distant supervision has been used to directly train a supervised model on the end task. Although models trained on distant supervision data improve general domain performances, they perform poorly in domain-specific evaluations due to multiple reasons:

1. **Annotation scheme** The training data needs to be engineered specifically to a desired annotation scheme and rules. In most cases, it is infeasible to exactly match the scheme of the test data, which gives a large advantage for in-domain gold datasets despite the size of the silver ones [5].
2. **In-domain data** Current models are predisposed towards high-frequency observations [122], where test data must come from the same distribution of train data in order to obtain the desired performances [159]. That is, training on small in-domain datasets is preferred on training on large out-of-domain ones.
3. **Noisy data** Commonly, NLP models are trained on high quality datasets, yet automatically annotated corpora are known to contain a portion of noise (in our case study at around 20%). The main reason behind this is that the heuristics for automatically annotating training data sometimes fails, which leads to noise. Previous studies [27, 43], show that the final performance of high capacity models (e.g. deep neural networks) is harmfully affected by datasets diluted with noisy examples.

The small amount of annotated data in NLP have tied word representations learning to unsupervised tasks like context prediction and language modeling, or supervised task with limited amount of data (like NMT). The main reason behind this is the absence of large-scale manually labelled data for most NLP tasks. Motivated by the great success of representation learning techniques, we were curious about the ability of learning useful representations from distant supervision data. Also, motivated by the limitations in using distant supervision data as training material, we propose to leverage distant supervision in order to learn word representations that can be further used as features. We argue that, if massive amount of labelled data is available, good representations could be learned on

downstream supervised tasks. To the best of our knowledge, this subject has not been yet explored.

Although, distant supervision is applied to a wide range of tasks (e.g. RE, NER, FGET), we choose NER as our end application for comparison and evaluation. We focus on NER because it is a fundamental and prerequisite for a wide range of high level NLP tasks like question answering and dialogue systems. Also, because most of prior works on data generation methodologies with distant supervision map, and evaluate on NER, this facilitates the comparison with previous works.

1.2 Contributions

This thesis presents an improvement of existing methods to generate annotated data from Wikipedia with distant supervision. We show its applications to NER using modern representation learning paradigms. The main contributions of this thesis are as follows:

- We revisit the idea of mining distant supervision data out of Wikipedia. We aim to detect the maximum number of annotations (hyperlinks) while introducing the minimum amount of noise. We propose rule-based labelling heuristics that are adapted to the specificity of Wikipedia. We created two variants of our data in order to support experiments on two downstream NLP tasks: WiNER for NER, and WiFiNE for fine-grained entity typing.
- We propose a cross-domain evaluation metric for NER, and compare performances between multiple approaches. Results show that feeding models with distant supervision annotations improves cross-domain performance, and that deep neural models benefit the most.
- We propose a simple yet efficient feature-based classifier to improve named entity classification. It computes 12 features computed over an arbitrary large part of distant supervision data.
- We evaluate the impact of WiFiNE as training material on the state-of-the-art fine-grained entity typing system on 2 manually annotated benchmarks (FIGER (GOLD) and ONTONOTES). Experiments show that training the system on

WiFiNE improves the performances on both benchmarks compared by the one trained on previously proposed distant supervision data, leading to state-of-art performances.

- We further propose to embed words and entity types into a low dimensional vector space, we train from annotated data produced by distant supervision. By doing so we can learn, for each word, a vector that encodes the similarity with entity types. Adding this vector as features into a vanilla RNN model leads to state-of-the-art performances on 2 standard NER benchmarks.
- We describe a special type of deep contextualized word representation that is *learned from* distant supervision annotations and *dedicated to* named entity recognition. Our extensive experiments on 7 datasets show systematic gains across all domains over strong baselines, and demonstrate that our representation is complementary to previously proposed ones. Also, we report new state-of-the-art results on 2 standard NER benchmarks.
- Wikipedia automatically annotated corpora, pre-trained word representations, and source code developed during this thesis are made publicly available and can be downloaded from
<http://rali.iro.umontreal.ca/rali/en/wikipedia-ds-cont-emb>.

1.3 Previously Published Material

Distant supervision annotated data produced in this thesis are built on top of the work that was done during my master [51]. Two articles - preliminary works to this thesis - that study coreference phenomena in Wikipedia where then published in conference proceedings:

- Abbas Ghaddar and Philippe Langlais. WikiCoref: An English Coreference-annotated Corpus of Wikipedia Articles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Portorož, Slovenia, 05/2016 2016

- Abbas Ghaddar and Phillippe Langlais. Coreference in Wikipedia: Main concept resolution. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 229–238, 2016

The research documented in this thesis has been published in three conference proceedings and one workshop:

1. Abbas Ghaddar and Phillippe Langlais. Winer: A wikipedia annotated corpus for named entity recognition. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 413–422, 2017
2. 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
3. Abbas Ghaddar and Phillippe Langlais. Robust Lexical Features for Improved Neural Network Named-Entity Recognition. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1896–1907, 2018
4. Abbas Ghaddar and Philippe Langlais. Contextualized Word Representations from Distant Supervision with and for NER. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 101–108, 2019

(55 total citations as for 31/12/2019)

1.4 Thesis Structure

- Chapter 2 gives an overview on state of the art methods for representation learning in NLP.
- Chapter 3 presents the task of Named Entity Recognition (NER), and review approaches, resources and applications of distant supervision for NER.
- Chapter 4 describes our methods to generate distant supervision data out of Wikipedia, and the creation of two corpora to support entity typing tasks (Article 1 and 2).

- Chapter 5 evaluates the annotation quality intrinsically on a manually labeled set of mentions) and extrinsically by using the corpus as training data for: named-entity recognition and fine-grained entity typing (Article 1 and 2).
- Chapter 6 presents our approach to induce low-dimensional lexical features and its application to NER (Article 3).
- Chapter 7 explores the idea of generating a contextualized word representation from entity type distant supervision annotations (Article 4).
- Chapter 8 summarizes the work of this thesis and suggests future work directions.

CHAPTER 2

REPRESENTATION LEARNING

For roughly two decades, natural language processing tasks have been tackled by number of hard-coded features that were known to correlate well with a given task, which in turn are fed to a machine learning algorithm. The goal of Deep Learning (DL) is to learn multiple layers of representations, or features, of increasing complexity and abstraction [59]. One important field of DL for NLP is word representation learning, where the goal is to develop algorithms that learn (in an unsupervised manner) general, low dimensional representations of word (called embeddings). The ultimate dream of such approaches is to be able to replace hand-engineered features in supervised NLP tasks. This chapter review main approaches, and the evolution of representation learning in the last decade.

2.1 Word embeddings

Word representations [103, 118, 157], also known as word embeddings, are feature vectors generated using unsupervised training on large quantities of unlabelled text. Word representations encode useful semantic and syntactic information about individual words into a low-dimensional space. Word representations are based on the assumption that words sharing similar neighbors tend to have similar representations. They are considered as a key element in multiple NLP tasks. In entity classification, it has been shown that entities with the same type have similar word representations, which makes them useful features. For example, company names like *Google*, *Microsoft*, *Facebook* will hopefully have similar word representations, as will do person names like *Obama*, *Clinton*, and *Trump*.

In an early stage, Bengio et al. [16] proposed an interesting architecture of Neural Language Model (NLM) to learn jointly a language model and word representations. The main motivation behind this work was to replace the one-hot encoding (a

feature vector that has the same length as the size of the vocabulary) of words by a low-dimensional continuous vector as a solution of *the curse of dimensionality* [15]. This work has attracted the neural network community to develop more efficient methods for neural word embeddings [32, 33, 102, 157].

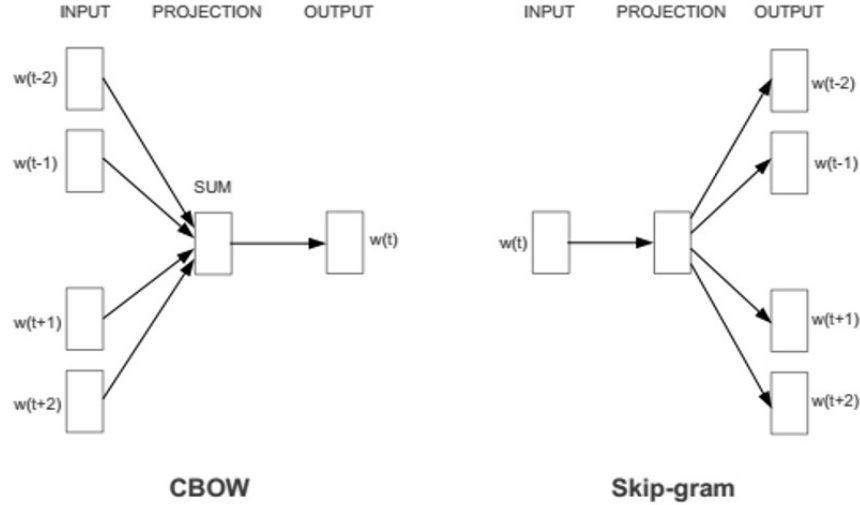


Image: Wang et al., 2015

Figure 2.1 – Illustration of the CBOW and the Skip-gram models proposed by Mikolov et al. [104]. Source [104]

Since 2010, two popular approaches for learning word embeddings dominated the field: *word2vec* [104] and *glove* [118]. In *word2vec*, the authors proposed two efficient models to formulate the task: *CBOW* and *Skip-gram* (Left and right sides of Figure 2.1). Given a sentence $s = (w_1, \dots, w_n)$, the first task consists in predicting the word (w_t) given its context (w_{t-2}^{t+2}), while *Skip-gram* goal is to predict the context given a word. For a target word at position t and its context at position c , the binary logistic loss for *Skip-gram* model is computed as follows:

$$l = \log \left(1 + e^{-s(w_t, w_c)} \right) + \sum_{n \in \mathcal{N}_{t,c}} \log \left(1 + e^{s(w_t, n)} \right),$$

where $s(w_t, w_c) = \mathbf{u}_{w_t}^\top \mathbf{v}_{w_c}$ is the scoring function between the target word and its context. \mathbf{u}_{w_t} and \mathbf{v}_{w_c} , are the trainable word embeddings vectors corresponding to w_t

and w_c respectively. $\mathcal{N}_{i,c}$ is a set of negative examples sampled from the vocabulary used to estimate the negative log-likelihood. Furthermore, Mikolov et al. [103] show that semantic relations can be extracted with simple arithmetic over the learned vectors. Figure 2.2 shows that words with similar meaning (country, capital) end up laying close in the vector space. For example, we can induce the capital of Germany using analogies over word vector with: $France - Paris + Germany \approx Berlin$.

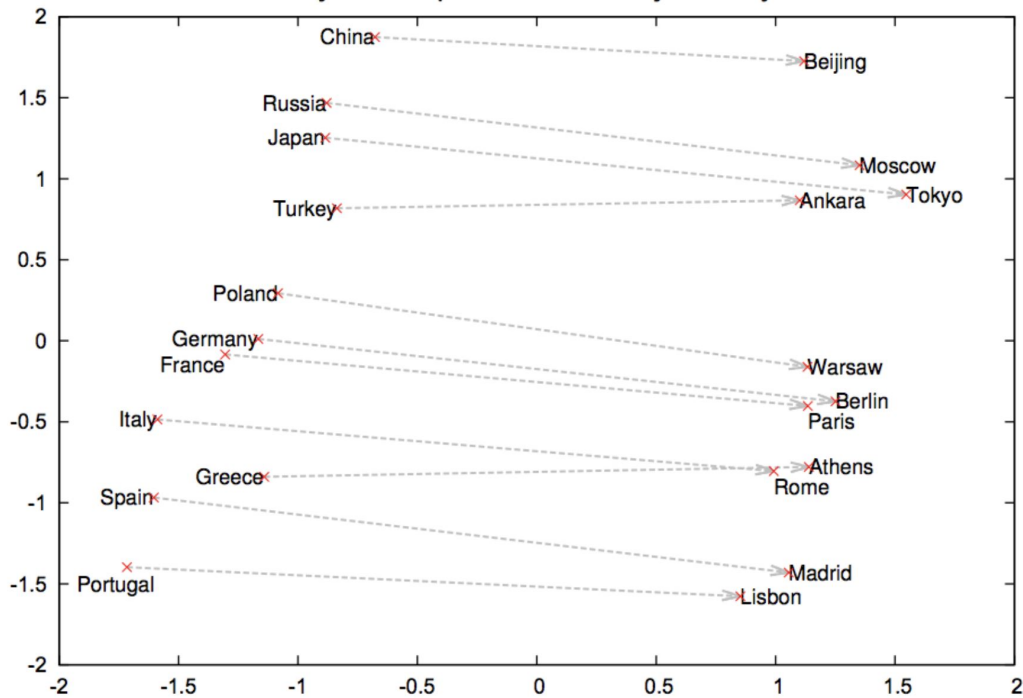


Figure 2.2 – Two-dimensional representation of the vector space of *word2vec* embeddings of selected cities and their respective capitals. Source [104]

Pennington et al. [118] argues that *word2vec* embeddings omit global information (corpus statistics). The authors propose an entirely different starting points to calculate words embeddings that capture both local and global information, leading to the so called *Glove* embeddings. The approach leverages the matrix of word-word co-occurrence counts in order to represent semantic relations between words.

Figure 2.3 shows co-occurrence probabilities and ratios between two target words (*ice* and *steam* and 4 selected context words. For example $P(solid|ice)$ is the probability that the word *solid* appears in the context of word *ice*. The basic idea behind *Glove* is

Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k ice)/P(k steam)$	8.9	8.5×10^{-2}	1.36	0.96

Figure 2.3 – word-word co-occurrence probabilities and ratio between the target words (*ice* and *steam*) and 4 selected context words. Source [118].

that the co-occurrence ratios between two words in a context are strongly connected to meaning. For example, we can observe that the nature relation ($nature_of(ice, solid)$ and $nature_of(steam, gas)$) can be extracted directly from the ratio probability. The ratio $P(solid|ice)/P(steam|gas)$ is large (or small in the other sense) while the ratio of words that are related (e.g. *water*) or unrelated (e.g. *fashion*) to both target words is near one. In a nutshell, the objective in *Glove* is to learn word vectors such that their dot product equals the logarithm of words probability of co-occurrence. Given words i and j , the objective is to minimize the weighted least squares regression loss:

$$l = f(X_{ij})(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij})^2 \quad (2.1)$$

$f(x) = \min(1, x/x^{\frac{3}{4}})$ is a weighting function, X_{ij} is the number of times word j occurs in the context of word i . w_i and \tilde{w}_j are word vectors to learn for words i and j respectively, while b_i and \tilde{b}_j are bias terms.

Glove and WORD2VEC are still considered as an essential component and are widely used as input features in state of the art systems for Semantic Role Labeling [61, 152], Coreference Resolution [83, 84], or Entity Linking [80, 144]. Nevertheless, there have been several endeavors to extend previous methods in order to enrich word embeddings and overcome some of their shortcomings.

Levy and Goldberg [85] generalize the *Skip-gram* model to support arbitrary syntactic context rather than a fixed-size windows context. First, they automatically parsed the corpus with dependency labels as in Figure 2.4, and the context of a target word is the set of words which modify the target coupled with their corresponding relations. For ex-

ample, the context for the word *discoveries* is: *scientist/nsubj, star/dobj, telescope/prep-with*, compared with the original *word2vec* model (window of 2): *Australian, scientist, star* and *with*. Dependency-based contexts allow the model to avoid accidental target-context words (*discoveries,Australian*), while capturing important words at any distance (*discoveries,telescope*) without the need to use a large window size that introduces noisy context.

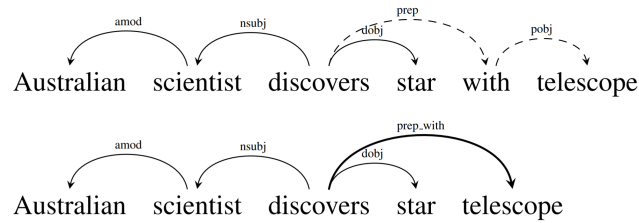


Figure 2.4 – Example of a sentence with dependency labels used in the work of Levy and Goldberg [85]. Source [85]

Neelakantan et al. [108] present an extension to the *Skip-gram* model that learns separate vectors for each word sense to better represent linguistic phenomenon like polysemy.

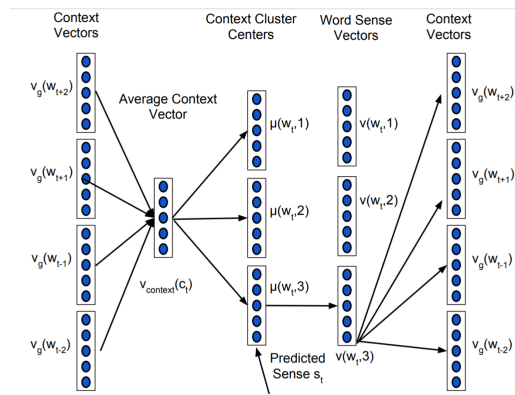


Figure 2.5 – Sentences with dependency labels from Neelakantan et al. [108], where parse tree is used to obtain arbitrary context words. Source [108]

Figure 2.5 illustrates the architecture of the Multi-Sense Skip-gram (MSSG) model, given a word at position t word sense embedding is calculated as follows:

1. the context vectors $c_t = \{w_{t-k}, \dots, w_{t-1}, w_{t+1}, \dots, w_{t+k}\}$ (k equals 2 in our example) are averaged into $(v_{context}(c_t))$.
2. the authors use cosine similarity between $v_{context}(c_t)$ and context cluster centers to determine the nearest cluster $\mu(w_t, i)$ and consequently the corresponding word sense embedding $v(w_t, i)$ ($i = 3$ in the Figure 2.5).
3. Only the vector of the selected word sense is used to predict context vectors c_t .

In FastText [19], the authors propose to enrich the original *Skip-gram* model of [104] by sub-words information. Each word is represented by the sum of its *n-grams* character embeddings in order to better model rare words.

2.2 Contextualized Word Representation

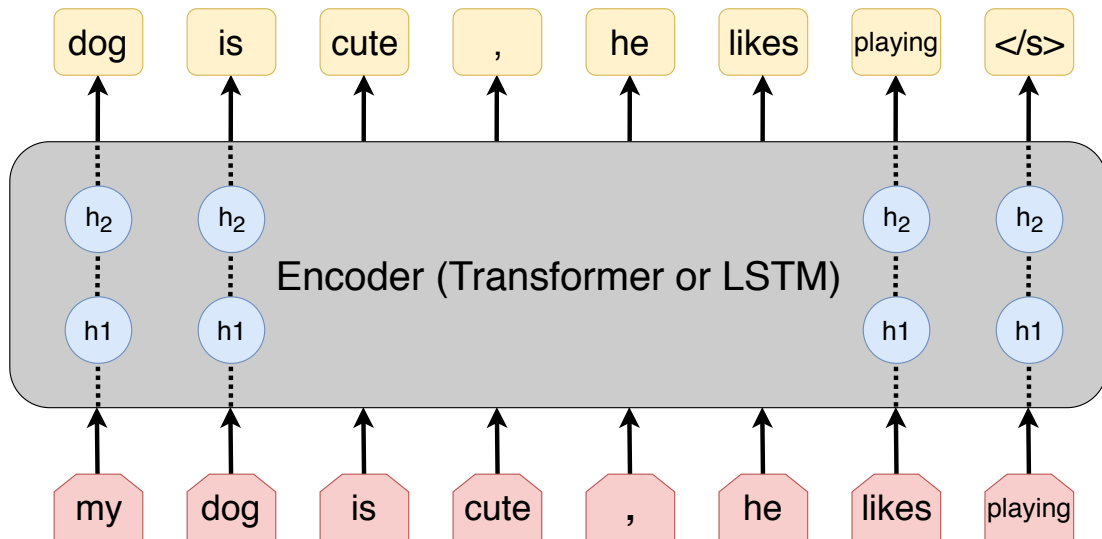


Figure 2.6 – The general framework of language model pre-training. Given a sentence, the goal is to predict, at each step, the next word given the previous context. A sequence encoder such as LSTM or Transformer is employed to encode syntactic and semantic features. The hidden state are considered as *contextualized word representations* because they varies for each word depending on the context.

Pre-trained word embeddings have been shown useful in NLP tasks, but cannot provide information about the exact sense in a particular context. For instance, with pre-trained embeddings, the word *France* will be initialized with the same –country like–

embedding in both input sentences: « *France is a developed country* » and « *Anatole France began his literary career* ». Depending on previous and future contexts, the embedding of *France* in the latter example should resemble to a given name.

In the last 2 years, several endeavors have attempted to learn context-dependent representations. One approach consists in training an encoder for a large NLP task (language modeling in Figure 2.6), and transferring the learned representations to another supervised task. Previous works [119, 120] has explored using language models in addition to word embeddings with positive results. The approach consists in learning an unsupervised deep language model on a large text corpus. Figure 2.7 illustrates a commonly used architecture for language modeling: 2 LSTM [63] layers [71]. The unsupervised task consists in predicting the next word given its previous context. The chain rule is used to model joint probabilities over word sequences:

$$p(w_1, \dots, w_n) = \prod_{i=1}^N p(w_i | w_1..w_{i-1})$$

The context of all previous words is encoded with an LSTM, and the probability over words is predicted using a Softmax (or NCE) over the output layer. The representations (red rectangle in Figure 2.7) at the input layer and hidden LSTM layers capture context-dependent aspects of word meaning using future (forward LM) and previous (backward LM) context words. After training the model, LM internal states can be used to generate representations that vary across linguistic contexts. The context-based representation of each word is a function of all words in the sentence. This approach produces a rich syntactic and semantic word representation [14], and can handle the limitations of traditional embeddings, among which is polysemy.

To add contextual embeddings to NLP downstream tasks (Figure 2.7), we first run the encoder on the input sentence and then concatenate the internal states and pass the enhanced representation into the task specific supervision. In our example, the word *France* will get a « last name » representation because family names will get high probability by using either forward LM ($p(X = \text{last_name} | \text{Anatole})$), or backward LM

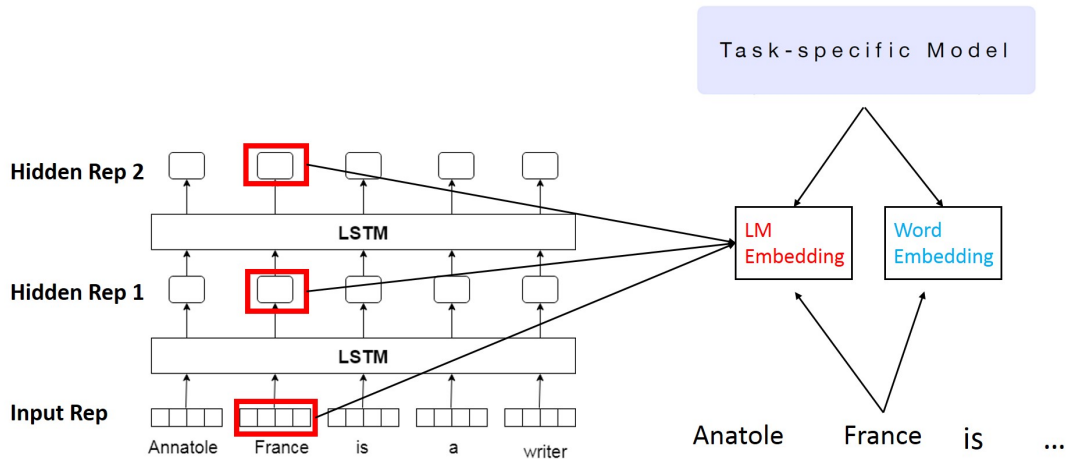


Figure 2.7 – Context-based representations generated by a neural language model for an input sentence of a specific supervised task (NER, QA,...).

$(p(X = \text{writer_last_name} | \text{writer a is}))$. Peters et al. [120] show significant improvements (Table 2.I) over the state of the art across six challenging NLP problems using LM embeddings (called ELMo). Due to its effectiveness, ELMo became a default choice for NLP applications, where plugging ELMo into an existing system typically leads to better accuracy.

TASK	PREVIOUS SOTA		BASELINE	ELMO + BASELINE	INCREASE (ABSOLUTE/RELATIVE)
SQuAD	Liu et al. [94]	84.4	81.1	85.8	4.7 / 24.9%
SNLI	Chen et al. [26]	88.6	88.0	88.7	0.7 / 5.8%
SRL	He et al. [61]	81.7	81.4	84.6	3.2 / 17.2%
Coref	Lee et al. [83]	67.2	67.2	70.4	3.2 / 9.8%
NER	Peters et al. [119]	91.9	90.1	92.2	2.1 / 21%
SST-5	McCann et al. [99]	53.7	51.4	54.7	3.3 / 6.8%

Table 2.I – Gains over state-of-the-art models (at that time) by adding ELMo [120] as feature to single model baselines across six benchmark NLP tasks: Reading Comprehension (SQuAD), Textual Entailment(SNLI), Semantic Role Labeling (SRL), Coreference Resolution (Coref), Named Entity Recognition (NER), and Sentiment Analysis (SST-5).Source [120]

Taking advantage of the abundance of machine translation data, one can generate contextualized embeddings using a supervised task. The approach consists in using rep-

representations produced by neural machine translation encoders. In CoVe [99], the authors trained an English-to-German machine translation system (left side of Figure 2.8), and use the internal states of the encoder as input for supervised tasks (right side of Figure 2.8). Similarly, Conneau et al. [35] used the internal states of neural models pre-trained on large Natural Language Inference datasets as features vectors for downstream tasks.

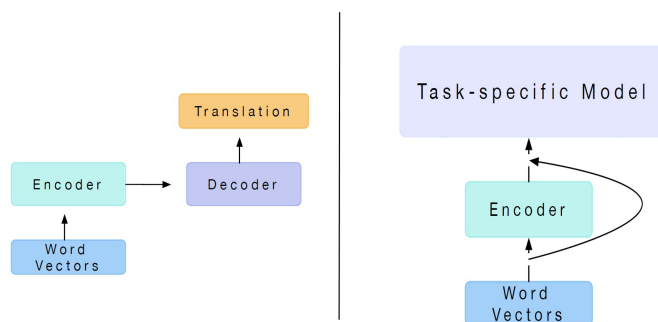


Figure 2.8 – Left Figure: An encoder decoder neural machine translation model is trained offline. Right Figure: the encoder is used to obtain representations, which in turn are used as input for supervised tasks. Source [99]

Peters et al. [121] study the impact of the language model architecture (e.g. LSTM [71], CNN [39], or Transformer [160]) on the end performance on four NLP tasks, and the properties of representations learned by each architecture. While all architectures produce high quality representations, empirical results show that downstream tasks benefits the most from contextual information driven from the bi-LSTM LM. The representation of each token is the concatenation of representations crafted from left-to-right and right-to-left language models. Due its sequential nature, the biLM suffers from slow training and inference time, which limits the possibility to scale up model size and data regime. The authors suggest that a very large model based on a computationally efficient architecture (e.g. transformer) and trained on massive amount of data could further improve the results.

2.3 Fine tuning Approaches

Transferring the knowledge from pre-trained LM to downstream supervised tasks in the form of additional feature vectors has proven to be effective, but one still needs to design a specific model architecture for each task. Therefore, recent researches have looked to push forward this approach by designing a *fine-tunable pre-trained* model. That is, the pre-trained language model is fine-tuned for a supervised downstream task, thus few parameters are learned from scratch. Howard and Ruder [67] proposes a universal LSTM-based language model that can be fine-tuned for text classification tasks. OpenAI GPT[127] is a Transformer-based fine-tunable language model that improves upon the state of the art in 9 NLP tasks including: commonsense reasoning, question answering, and textual entailment.

At the end of 2018, a group of researchers from Google released BERT [41], which was considered by many as a *game-changer* in NLP. BERT (**B**idirectional **E**ncoder **R**epresentations from **T**ransformers) builds on top of previously proposed methods for self-training to overcome the limitation (unidirectionality) of a standard language model objective. Transformer [160] is a multi-layer bidirectional sequence encoder that was originally designed for machine translation. A single Transformer layer is build upon feed forward neural networks, residual connections and an attention mechanism (see the original paper for a detailed description).

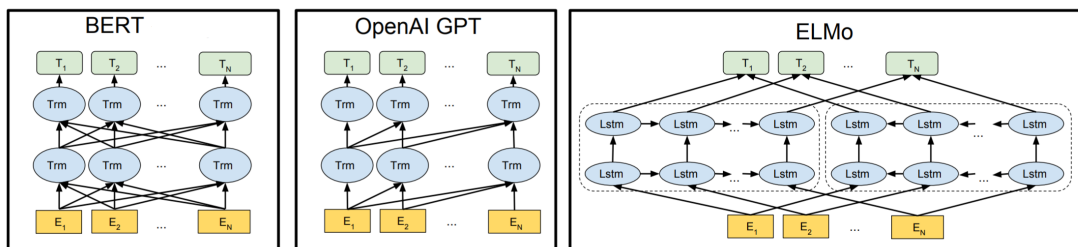


Figure 2.9 – Key differences between BERT [41], OpenAI GPT [127] and Elmo[120] pre-training objectives. Source [41]

In BERT, the authors propose two novel tasks for pre-training: Masked Language Model and next sentence prediction. The goal of the first task is to predict masked

tokens (marked by [MASK] in Figure 2.10) in the input sequence, while the goal of the second task is to predict if sentence B is the actual next sentence of A. The authors randomly masked 15% of the word input, where a softmax on the vocabulary is used to predict masked tokens¹. For the next sentence prediction task, 50% of the time B is the actual next sentence of A, and 50% of the time it is a random sentence from the corpus.

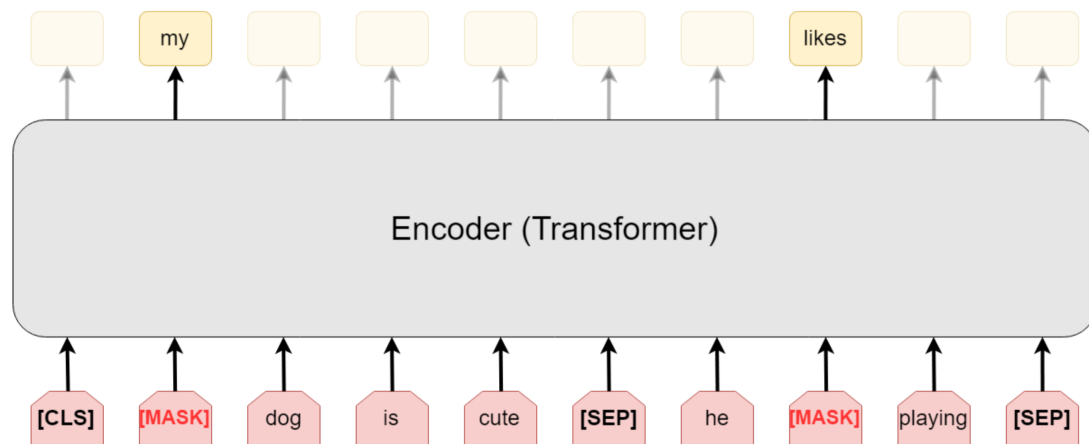


Figure 2.10 – Illustration of an input sequence of BERT [41]. Some words are randomly masked in the input sentence and the goal is to predict them at the output. No masked word are ignored in the loss calculation (transparent rectangle).

Both tasks enable the bidirectionality during pre-training (BERT in Fig. 2.9) compared with the left-to-right language model of (OpenAI GPT in Fig. 2.9), or the shallow concatenation of independently trained left-to-right and right-to-left language models (ELMo in Fig. 2.9). During pre-training, the input of BERT (Figure 2.10) consists of a special classification token [CLS], tokens of a sentence A, a special token separator [SEP], and the tokens of a sentence B. To improve the handling of rare and unknown words, the input sequence consists of WordPiece tokens [167].

As shown in Figure 2.11, BERT can easily be fine-tuned for a large range of sentence-level and token-level tasks including: single and sentence pair classification (e.g. sentiment analysis, textual entailment), where only a feed forward and a softmax layer (on top of the [CLS] token) are learned from scratch. Also, multiple choice selection tasks [177] can be framed as a binary classification of sentence pair using BERT. The paragraph and

1. No prediction is made for unmasked tokens.

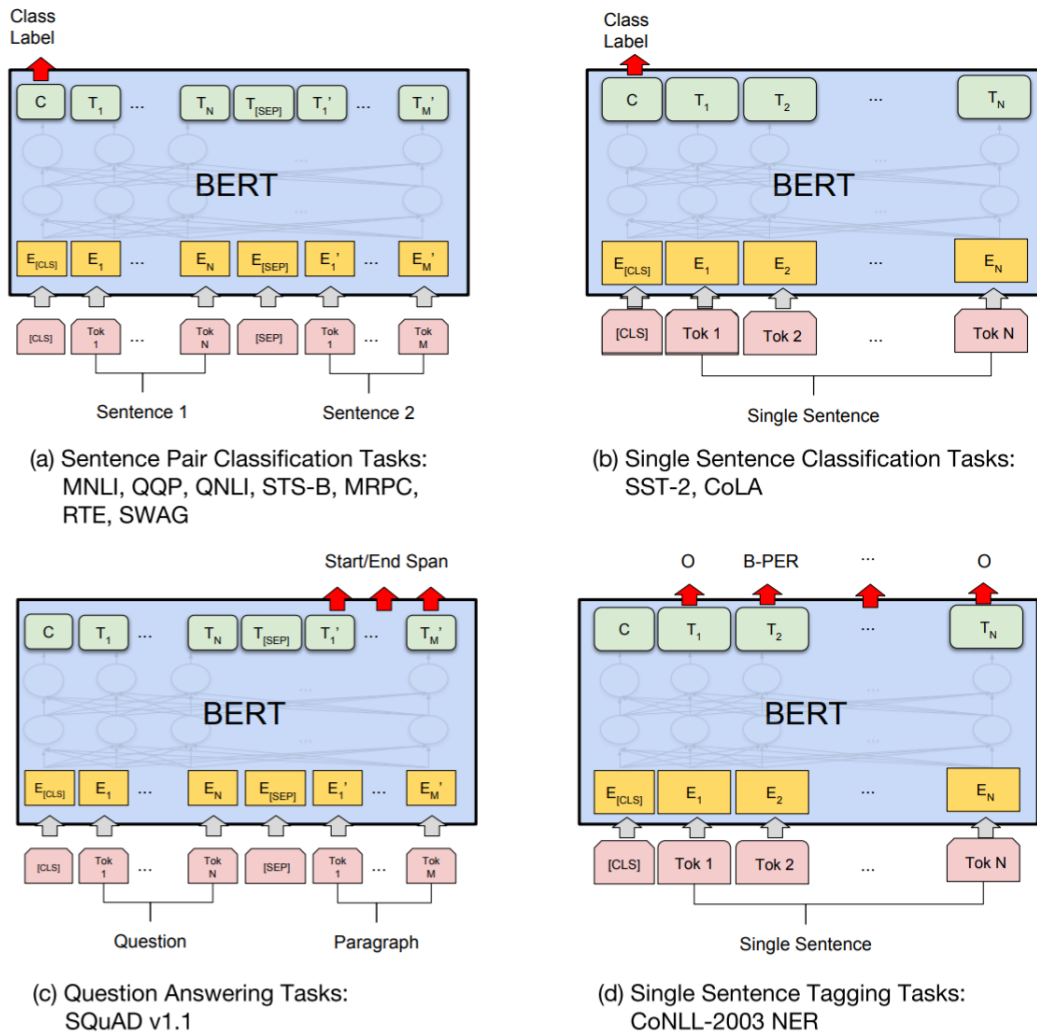


Figure 2.11 – Illustration of how BERT[41] can be fine tuned for a wide range of NLP tasks. Source [41]

answer are concatenated (separated by [SEP]) as input, where sequences with correct answer are treated as positive examples and those with wrong answers as negative. BERT support tasks extractive question answering [130], as well as sequential tagging tasks by predicting a label for each token in the input sequence. BERT improves state-of-the-art results on no less than eleven natural language processing tasks. Furthermore, BERT internal states can be used as a contextualized representation. The authors of BERT show for instance competitive results when BERT vectors are plugged as input features in a vanilla bi-LSTM model for NER.

Following the great success of BERT, a series of papers built on top of the original model to support new tasks and domains:

- spanBERT[70] for span-level tasks like coreference resolution and relation extraction.
- GlossBERT [68] for word sense disambiguation [107].
- Andor et al. [7] generalize BERT to perform lightweight numerical reasoning for reading comprehension.
- BioBERT [82] a special version of BERT to generate domain specific representations for biomedical NLP tasks.
- For information extraction related tasks like: passage re-ranking [109] and open-domain question answering [169].

In XLNet [170], the authors propose a new model that overcomes two of BERT main weaknesses: the fixed length context, and pre-train fine-tune mismatch due to the special [MASK] tokens. The model uses Transformer-XL [37], which adds a *memory cell* to the original Transformer architecture to handle very long sequences with a recursive mechanism. In addition, during pre-training the authors replaced the masked LM by a permutation LM task, where the model can be conditioned on an arbitrary sequence to predict the target token. For example, the model might be asked to calculate $P(x_3|x_5, x_7, x_1)$, while standard LM can only be conditioned on $P(x_3|x_{<3})$ or $P(x_3|x_{>3})$. Thus, the bidirectionality of the model is maintained without the need to corrupt the input sequence with the [MASK] tokens.

In RoBERTa [95], the authors have found that BERT was under-trained. They pro-

pose some modifications and better way to fine-tune the model that systematically improves the results. Modifications includes: **(1)** using 10 times more data (160GB) for pre-training; **(2)** removing the next sentence prediction task; **(3)** dropping short input sequences; **(4)** larger batch size and training steps; **(5)** tuning the optimizer hyper-parameters. The authors shows that carefully tuning the original BERT outperforms all models published so far.

In ALBERT [79], yet another variant of BERT, the main focus of the authors is on reducing the model size. They propose to share the parameters across the Transformer layers, and to project embeddings into a low dimensional space of size E , and then project it to the hidden space $H \gg E$ (factorized embedding). These modifications significantly reduce the number of parameters (by x10), which in turn increase the training speed of BERT and reduces memory consumption. In addition, the authors argue that randomly picking 2 segments made the next sentence prediction task easy and consequently ineffective. For generating negative examples, they swap 2 consecutive segments rather than picking them randomly from the corpus. The Inter-sentence coherence task, as they call it improves performances on downstream tasks.

2.4 Conclusion

This chapter gave an overview of the two main approaches for word representation learning: classic (fixed) embeddings, and contextualized representations. Also, we pass through fine-tunable language models (BERT) approaches that emerged as the *Swiss Army Knife* for NLP applications. We detailed the main methods for each approach, and reviewed other refinements to improve over them. While most of these works focus on learning representations from unlabeled text, we explore the usefulness of massive amount of automatically annotated data for classic and contextualized word representation learning.

CHAPTER 3

DISTANT SUPERVISION FOR NER

In this chapter, we first present the task of named entity recognition and the distant supervision methods for automatic data augmentation for this task.

3.1 Named Entity Recognition

Named Entity Recognition (NER) is the task of identifying textual mentions and classifying them into a predefined set of types. While NER task focuses on a small set of types (between 4 and 17), fine-grained entity typing deals with a larger type sets (between 89 and 112). The top part of Figure 3.1 shows an example of NER with coarse types like: PERSON, LOCATION, ORGANIZATION, while the bottom part shows the task with more fine grained types that are stored in a hierarchical structure.

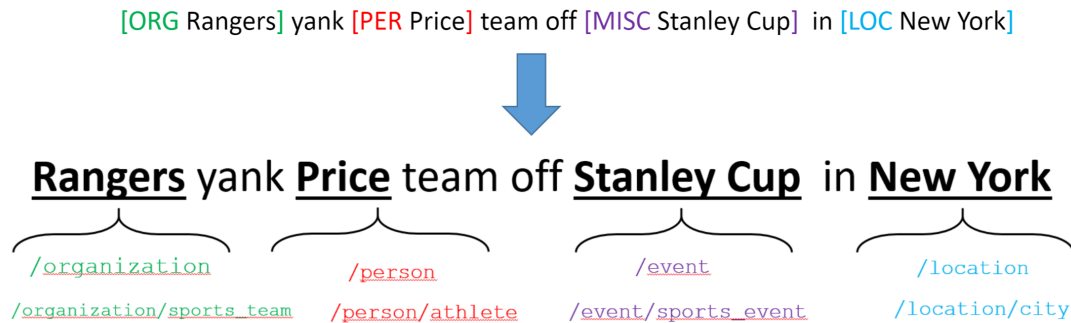


Figure 3.1 – An illustration of the named entity recognition task. Given a sentence, the goal is to identify entity mentions and to classify them into predefined categories. The top part shows NER with coarse categories, while the bottom part shows the task with more fine grained types.

NER plays a vital role in natural language understanding and fulfill lots of downstream applications, such as entity linking, relation extraction, coreference resolution and question answering. For example, knowing that an entity mention is a PERSON would:

1. prevent an entity linking system from linking that entity to a no person Wikipedia page.
2. limit the number of possible relations with other entities in relation extraction.
3. prevent coreference resolution system from merging that entity with a cluster of location entities.
4. be a great indicator for q question answering system of certain type of questions (e.g. *PERSON* entities are potential answer for *who* question).

Due its importance, the NER task is an active domain of research, where it is being investigated from dataset creation and feature engineering to modeling and evaluation.

3.2 Datasets

In the last two decades, named entity recognition imposed itself on the natural language processing community as an independent task in a series of evaluation campaigns such as MUC-1996 [60], MUC-1997 [29] and CoNLL-2003 [154]. This gave birth to various corpora designed in part to support training, adapting or evaluating named entity recognizers. Most of the aforementioned datasets were created on top of news documents, which limited the scope of applications of the technology. It is now widely accepted that NER systems trained on newswire data perform poorly when tested on other text genres [10, 111]. Thus, there is a crucial need for annotated material of more text genres and domains. This need has been partially fulfilled by some initiatives that manually created datasets in order to study NER on different domains, such as biomedical (I2B2[153]), social media (W-NUT [40, 150]), and finance (FIN [6]) domains.

Table 3.I summarizes the main characteristics of manually annotated NER datasets. CoNLL-2003 and ONTONOTES 5.0 are considered as the standard benchmarks for evaluating and comparing systems. The CoNLL-2003 NER dataset [154] is a well known collection of Reuters newswire articles that contains a large portion of sports news. It is annotated with four entity types: *Person* (PER), *Location* (LOC), *Organization* (ORG) and *Miscellaneous* (MISC). The four entity types are fairly evenly distributed, and the train/dev/test datasets present a similar type distribution. The ONTONOTES

Corpus	Year	Text Source	#Tags	URL
MUC-6[60]	1995	Wall Street Journal texts	7	https://catalog ldc.upenn.edu/LDC2003T13
MUC-6 Plus[28]	1995	Additional news to MUC-6	7	https://catalog ldc.upenn.edu/LDC96T10
MUC-7[29]	1997	New York Times news	7	https://catalog ldc.upenn.edu/LDC2001T02
CoNLL03[154]	2003	Reuters news	4	https://www.clips.uantwerpen.be/conll2003/ner/
ACE[42]	2000 - 2008	Transcripts, news	7	https://www ldc.upenn.edu/collaborations/past-projects/ace
OntoNotes[66]	2007 - 2012	Magazine, news, conversation, web	89	https://catalog ldc.upenn.edu/LDC2013T19
W-NUT[40, 150]	2015 - 2018	User-generated text	18	http://noisy-text.github.io
BBN[162]	2005	Wall Street Journal texts	64	https://catalog ldc.upenn.edu/ldc2005t33
NYT[140]	2008	New York Times texts	5	https://catalog ldc.upenn.edu/LDC2008T19
WikiGold [113]	2009	Wikipedia	4	https://figshare.com/articles/Learning_multilingual_named_entity_recognition_from_Wikipedia/5462500
WebPages [132]	2009	Web	4	https://cogcomp.seas.upenn.edu/page/resource_view/28
FIGER(GOLD)[91]	2012	Wikipedia	113	https://github.com/xiaoling/figer
N ³ [139]	2014	News	3	http://aksw.org/Projects/N3NERNEDNIF.html
GENIA[73]	2004	Biology and clinical texts	36	http://www.geniaproject.org/home
FSU-PRGE	2010	PubMed and MEDLINE	5	https://julielab.de/Resources/FSU_PRGE.html
BC5CDR[34]	2013	PubMed	3	http://bioc.sourceforge.net/
NCBI-Disease[44]	2014	PubMed	790	https://www.ncbi.nlm.nih.gov/CBBresearch/Dogan/DISEASE/
I2B2[153]	2015	Clinical Data	23	https://www.i2b2.org/NLP/DataSets
FIN[6]	2015	8k financial report	4	http://people.eng.unimelb.edu.au/tbaldwin/resources/finance-sec/
DFKI[141]	2018	Business news and social media	7	https://dfki-lt-re-group.bitbucket.io/product-corpus/

Table 3.I – List of manually annotated datasets for English NER, with the number of entity types (#Tags). source [86].

5.0 dataset [66, 124] includes texts from five different genres: broadcast conversation (200k), broadcast news (200k), magazine (120k), newswire (625k), and web data (300k). This dataset is annotated with 18 entity types, and is much larger than CoNLL.

3.3 Approaches to NER

In this thesis, we treat NER as a sequential labeling problem, which is the most commonly used approach in the literature. Figure 3.2 illustrates the high level architecture of a sequential tagger for NER. Each token at the input sequence is represented by a set of features. Based on these features, many machine/deep learning algorithms have been proposed to learn a model to produce a sequence of labels that represent the entity type assigned to each input token.

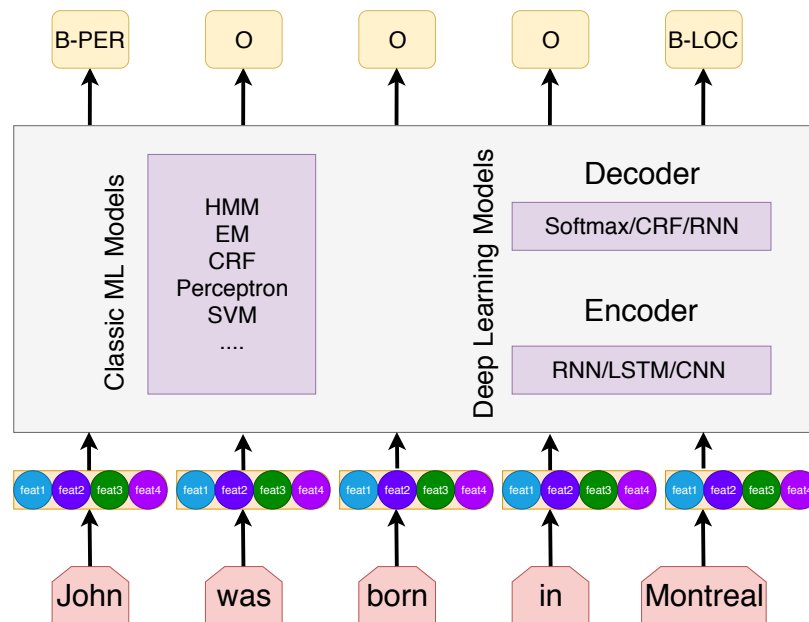


Figure 3.2 – High level architecture for NER model as a sequence labeling problem. Each word in the input is represented by a set of features. These features are fed into a classifier (gray box) which in turn produces a label per token at the output layer indicating the entity type of the token.

Traditional approaches to NER rely on heavy feature engineering coupled with classic machine learning algorithm like CRF [49, 50], SVM [87] and perceptron [132] as shown in Figure 3.2. Feature design is crucial for NER performance, especially when classic machine learning algorithms are used. Literature is fruitful with a wide range of local and global features that encode syntactic and semantic characteristics of words. We list the most commonly used word-level features by traditional approaches to NER:

- One hot encoding of the current, previous and next word n -grams.
- Brown clusters [23] (as a binary feature) of each word. The Brown algorithm assigns words with similar contexts (e.g. *Friday* and *Tuesday*) to the same cluster.
- Gazetteers are binary features that encodes the presence of word n -grams in a predefined lists of NEs such as lists of known persons, locations and organizations. These features are used in traditional [132] as well as in deep learning [30] approaches.
- Capitalization, also known as word shape features, characterize certain categories of capitalization patterns such as *allUpper*, *allLower*, *upperFirst*, *upperNotFirst*, *numeric* or *noAlphaNum* features.
- Part-of-speech tags of the current word as well as previous and next n -grams tags (usually $n \leq 3$).
- List of prefix and suffix character n -grams, which are strong predictors of the syntactic function of the word.
- Also, some works [76] assign document topics (e.g with topic modeling [18]) as features for each word.

On the other hand, deep learning approaches focus on a small set of features that are fed into neural networks encoder followed by a tag decoder. Popular approach to NER such as the systems [30, 119] use Bi-LSTMs and Convolutional Neural Networks (CNNs) as a sequence encoder, along with a CRF decoder. CNNs are used to encode character-level features (prefix and suffix), while LSTM is used to encode word-level features. Finally, a CRF is placed on top of those models in order to decode the best tag sequence. Pre-trained embeddings obtained by unsupervised learning are core features of those models. The capability of deep learning models to learn directly from the data, and the advancement in representation learning techniques has made NER systems to rely more on these features that includes:

- Word embeddings such as *word2vec*[104] and *Glove* [118].
- Deep contextualized word representation from language models like ELMo [120], GPT [127], FLAIR [4] and BERT [41].
- Character-level embeddings [78, 98] which are randomly initialized and learned

during training.

3.4 Metrics

In evaluation, we need to compare the true set of entities produced by human annotators, the predicted set of entities. In the literature, NER is traditionally evaluated in terms of precision (P), a measure of exactness, and recall (R), a measure of completeness, and the F-score corresponds to their harmonic mean:

$$\text{Precision} = \frac{TP}{TP + FP} \quad \text{Recall} = \frac{TP}{TP + FN}$$

$$\text{F-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

where TP (True Positive) are entities or tokens that are recognized by the system and match ground truth; FP (False Positive) are those that are recognized by the system and do not match ground truth; and FN (False Negative) are entities in the ground truth that are not recognized by the system.

When comparing results on standard benchmarks, the F1 score at the entity level [154] is commonly used. A more loose metric, F1 score at the token level is mostly used in out-of-domain evaluations due to annotation scheme mismatch [132]. At the entity level, the system receives a full score if it correctly predicts the segment and label of a given entity otherwise zero. At token level, the score is per token, that is, if the entity is composed of 3 tokens and 2 of them are predicted correctly the F1 score is 0.67. For an in-depth result analysis, some papers report per-class, macro- and micro-averaged F-scores. The macro-averaged F-score treats all entity types equally by computing the F-score independently for each class. In micro score all entities are treated equally by aggregating the contributions of entities from all classes.

Table 3.II shows the entity level F1 score on test sets of CONLL-2003 and ONTONOTES 5.0 for a wide range of approaches for NER. The top part of the table list feature-based supervised learning approaches for NER, while the bottom part lists the deep neural net-

Model	CoNLL	Ontonotes
Linear CRF [49]	86.86	82.48
Averaged Perceptron [132]	90.88	83.45
Joint Model NER, CR, EL [45]	-	84.04
Joint Model NER and EL [97]	91.20	-
Multi task learning [33]	89.56	-
BiLSTM-CRF [98]	91.21	-
BiLSTM-Char-CRF [30]	91.60	86.28
Iterated Dilated CNN [151]	90.54	86.99
LM-LSTM-CRF [93]	91.71	-
BiLSTM-CRF+LM [119]	92.20	-
BiLSTM-CRF+ELMo [120]	92.20	-
CVT + Multi-Task [31]	92.61	88.81
BERT Large [41]	92.80	-
Flair embeddings [4]	93.01	89.71
Shared LSTM [96]	92.60	-
CNN Large + fine-tune [11]	93.50	-

Table 3.II – Entity level F1 scores on test sets of CONLL-2003 and ONTONOTES 5.0 respectively. The model description show the main contribution of the paper. The first set are feature-based (classic) models; the second set are neural models without external knowledge as feature; the last set are neural models with external knowledge.

works systems. 93.50 and 89.71 are the state-of-the-art performances on CONLL-2003 and ONTONOTES 5.0 respectively at date of writing the thesis. However, 89.71 for in-domain evaluation is not considered as strong result, which challenge the robustness of current state-of-the-art models in real world application. We further detail this issue in Chapter 5.

3.5 Distant Supervision for NER

One major drawback of NER (and other NLP tasks) is the absence of a large-scale manually labeled data for most domains, which limits the use of NER in real world applications. Distant Supervision (DS) techniques are very promising as they can be used to overcome the lack of large-scale labelled data for NER. This technique mainly consists in generating training data out of partly annotated examples (e.g. Wikipedia)

that are linked to a knowledge base (such as Freebase). Surface form of links that point to knowledge base entries are considered the main source of annotation.

In NER distant supervision, human-created hyperlinks within web pages (Section 3.5.1) or encyclopedias entries like Wikipedia (Section 3.5.2) are considered the main sources of annotations. Despite being larger, web pages are considered noisier, less structured, and contains fewer links per page compared to Wikipedia articles. In addition, Wikipedia has the advantage of being supported by structured knowledge bases like Freebase [21], Yago [72] and Wikidata [161]. In this thesis, we choose Wikipedia as our source of annotation, following pioneer works on distant supervision for NER, thus facilitating comparisons.

3.5.1 Web page based Corpora

In 2012, Singh et al. [145] released the Wikilinks corpus which consists of non-Wikipedia web pages that contain links to English Wikipedia. First they crawl the web to discover these, then they apply several filters to ensure quality annotations, including: (1) Discarding web pages if they overlap significantly (>70%) with a single Wikipedia page; (2) Discarding all links that appear in tables, near images, or in obvious boilerplate material; (3) Keep a link if its anchor has a common token with either the Wikipedia page title or with one of its name variants (alias, redirect). Wikilinks¹ gathers roughly eleven million web pages that incorporate over 40 millions links to Wikipedia. According to authors, the corpus can be used for within/cross document coreference, entity linking, entity tagging among other tasks. Similarly to Wikilinks, Google released a version of the English-language web pages from Clueweb12 [47] automatically annotated with Freebase entities. While the annotation procedure is not described in the documentation, the team claims that the corpus contains high precision but low recall mentions. The corpus² comprises around 456 million web documents that contains at least one entity, annotated with over 6 billion mentions. These distant supervision generated corpora have been employed in a number of tasks including entity linking [92, 165]; and question

1. <http://code.google.com/p/wiki-links>

2. <http://lemurproject.org/clueweb12/FACCl/>

answering [133, 171].

3.5.2 Wikipedia for NER

Wikipedia is a large, multilingual, highly structured, multi-domain encyclopedia, providing an increasingly large wealth of knowledge. It is known to contain well-formed, grammatical and meaningful sentences, compared to say, ordinary internet documents. It is therefore a resource of choice in many NLP systems, see [101] for a review of some pioneering works. The English version, as of 13 April 2013, contains 3,538,366 articles thus providing a large coverage knowledge resource.

Redirect

- Obama
- 2008 Democratic Presidential Nominee
- 44th President of the United States
- 44th president of the united states of america
- B. H. Obama
- B. Hussein Obama
- B. Obama
- BARACK OBAMA
- BHOII
- Bacak Obama
- Barac Obama
- Barac obama
- Barach Obama
- Barack

Infobox

Personal details	
Born	Barack Hussein Obama II August 4, 1961 (age 54) Honolulu, Hawaii, U.S.
Nationality	American
Political party	Democratic
Spouse(s)	Michelle Robinson (m. 1992)
Children	Malia (b. 1998) Sasha (b. 2001)
Residence	White House
Education	Punahou School
Alma mater	Occidental College Columbia University (B.A.) Harvard Law School (J.D.)
Religion	Protestantism (see details) ^[1]
Signature	
Website	barackobama.com 

In June 1989, Obama met [Michelle Robinson](#) when he was employed as a summer associate at the Chicago law firm of [Sidley Austin](#).^[380]

Label

Wiki Article

Figure 3.3 – Excerpt from the Wikipedia article *Barack Obama*

An entry in Wikipedia provides information about the concept it mainly describes.

A Wikipedia page has a number of useful reference features, such as **internal links (hyperlinks)** which link a surface form (*Label* in figure 3.3) into other articles (*Wiki Article* in figure 3.3) in Wikipedia); **redirects** which consist in misspelling and names variations of the article title; **infoboxes** that are structured information about the concept being described in the page; and **categories** which is a semantic network classification. The aim of Freebase [20] was to structure human knowledge into a scalable tuple database, by collecting structured data from the web, where Wikipedia structured data (infoboxes) forms the skeleton of Freebase. As a result, each Wikipedia article has an equivalent page in Freebase, which contains well structured attributes related to the topic being described. Figure 3.4 shows some structured data from the Freebase page of *Barack Obama*.

Alias

Topic /common/topic

Also known as /common/topic/alias

Also known as

- Barack Hussein Obama, Jr.
- Barack Hussein Obama
- Obama
- President Obama
- Barack H. Obama II
- Barack Hussein Obama II
- Barack Obama II
- President Barack Hussein Obama II
- Sen. Barack Obama
- Barak Obama

90 values total >

Attributes: gender, type, profession,...

Notable for /common/topic/notable_for

US President

Country of nationality /people/person/nationality

United States of America

Gender /people/person/gender

Male

Profession /people/person/profession

Profession

- Politician
- Lawyer
- Writer
- Author
- Law professor

Lots of Links with other pages

Appointer /people/appointer

Appointment made /people/appointer/appointment_made

Appointed Role

- U.S. Global AIDS Coordinator
- Under Secretary of State for Public Diplomacy and Public Affairs
- Ambassador-at-Large for Global Women's Issues

264 values total >

Figure 3.4 – Excerpt of the Freebase page of *Barack Obama*

Transforming Wikipedia into a corpus of named entities annotated with entity types is a task that received attention in a monolingual setting [112, 155], as well as in a multilingual one [5, 137]. Because only a tiny portion of texts in Wikipedia are anchored, some strategies are typically needed to infer more annotations. Such a process typically yields a noisy corpus for which filtering is required.

Wikipedia articles:



Holden is an **Australian** automaker based in **Port Melbourne, Victoria**. The company was originally independent, but since 1931 has been a subsidiary of **General Motors** (GM). Holden has taken charge of vehicle operations for GM in **Australasia** and, on

Sentences with links:

Holden|**Holden** is an **Australian**|**Australia** automaker based in **Port_Melbourne,_Victoria**|**Port_Melbourne,_Victoria**.



NE-tagged sentences:

[**ORG Holden**] is an [**LOC Australian**] automaker based in [**LOC Port Melbourne, Victoria**].

Adjusted annotations:

[**ORG Holden**] is an [**MISC Australian**] automaker based in [**LOC Port Melbourne**], [**LOC Victoria**].

Figure 3.5 – Wikipedia to named-entity annotated corpus pipeline of Nothman et al. [112]. source [112]

Nothman et al. [112] describe an approach (Figure 3.5) that exploits links between articles in Wikipedia in order to produce named-entity mentions (person, location, organization, miscellaneous). They are making use of hand-crafted rules specific to Wikipedia, and a bootstrapping approach for identifying a subset of Wikipedia articles where the type of the entity can be predicted with confidence. Since anchored texts in Wikipedia lack coverage (in part because Wikipedia rules recommend that only the first mention of a given concept be anchored in a page), the authors also describe heuristics based on redirects to identify more named-entity mentions. They tested several variants of their corpus on three NER benchmarks and showed that systems trained on Wikipedia data may perform better than domain-specific systems in an out-domain

setting. The same technique was applied in [5, 114] on numerous foreign-language Wikipedia dumps, in order to generate named-entity annotations for these languages.

Ling and Weld [91] also used Wikipedia in the same manner to produce fine-grained entity type annotations. Each entry in Wikipedia that appears as an anchor link is assigned by types from its Freebase equivalent page, which in turn are mapped to a predefined entity tag set (113). The authors release their data, which has been used for training in a number of studies such as [135, 136, 143].

3.6 Enriching Wikipedia with Links

As discussed in the previous section, only a small set of hyperlinks are annotated due to the Wikipedia linking policy³. Previous works have proposed methods to identify missing links inside Wikipedia. In Noraset et al. [110], the authors tackle the task as an Entity Linking problem, and introduce *W3*, a classifier that identifies concept mentions in Wikipedia text using Wikipedia-specific features. Although the system produces high-precision links (0.98), it suffers from low recall (0.38). In average, *W3* find 7 new links in each Wikipedia article.

Another direction is the work of West et al. [164], which focuses on only adding links that improve the navigability of the resource. For example, *Flower* is an important concept to identify in the *Botany* article, but is useless in articles like *Wedding* or *Montreal*. They build their system on the top of logs of a Wikipedia-based human game that consists in finding the shortest path from a source to a target article.

Lastly, we introduce the method proposed by Raganato et al. [128] which is the most similar work to ours⁴. The authors use rule-based heuristics that are based on the structure of Wikipedia itself, and structured data from BabelNet (an ontology that combines various encyclopedias and dictionaries, e.g. WordNet and Wikipedia among others). The heuristics mainly consists in lexicalization matching of name variants and synonyms of concepts found in Wikipedia and BabelNet with potential missing links in Wikipedia pages. Also, they have heuristics to propagate verb, adjective and adverb

3. [http://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style_\(linking\)](http://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style_(linking))

4. We build on top and extends the heuristics proposed by the authors

senses extracted from WordNet. Their method increases by 3.5 times the number of links in Wikipedia, with a precision of 0.94 (measured on 1500 manually labelled example). The authors evaluated the usability of their corpus as a training material on different tasks (Entity Linking and Word Similarity).

Corpus	# Links	# Entities	# Documents	Link/Doc
Wikipedia (2014)	71.5	2.9	4.3	16.6
Wikilinks [145]	40.3	2.9	10.9	3.7
ClueWeb12 [47]	11240	5.1	1104	10.2
SWE [128]	162.6	4.0	4.3	37.8

Table 3.III – Comparison of different link-enriched corpora. Counts (# columns) are in millions.

Table 3.III summarizes the main characteristics of a number of existing automatically-annotated corpora. Although Wikilinks and ClueWeb12 corpora contain a large number of documents, they do suffer from low coverage and noise, compared to Wikipedia-based corpora. In the next Chapter, we describe a resource for NER that increases the number of links in Wikipedia without sacrificing much precision, for the sake of applications.

Our approach is based on the early observations made by Ghaddar and Langlais [52, 55] on the special characteristics of coreference phenomena in Wikipedia. The authors show that a Wikipedia-adapted coreference resolution approach can resolve the task with high accuracy. In this thesis, we study and develop a number of Wikipedia coreference labeling heuristics (Chapter 4) such as following out-links and out-links of out-links to enrich Wikipedia with hyperlinks. By mapping hyperlinks to their corresponding entity types we build two automatically labeled datasets WiFiNE and WiFiNE for NER and FGET receptively.

3.6.1 Recent works on DS

Despite the success of pre-trained language models for NER, they still need in-domain annotated data to predict NER labels (fine-tuning process). Consequently, many improvements can be made if large scale annotated data is provided to these strong mod-

els. Recently⁵, two studies [172, 178] have revisited distant supervision for NER. They describe a similar approaches to ours, except they incorporate deep learning in the annotation process. These works show the importance of the problem we tackle and reflect the increasing need for larger datasets in multiple domains. While they acknowledge the similarity to our work, they do not compare their resource to ours⁶.

DocRED [172] is a collection of Wikipedia introductions (first paragraphs) annotated with named entities and relations originally designed for document-level relation extraction [117, 126]. There are 2 versions of the dataset, one small manually annotated and a large one obtained via distant supervision [106]. To construct the dataset, more 107k documents were automatically annotated using spaCy [65] for NER and TagMe [48] for Entity Linking, Wikidata⁷ [46, 161] was used to infer relations between entities. Then, a randomly picked subset of 5k documents were manually corrected by human annotators.

AnchorNER [178], is yet another Wikipedia-based corpus automatically annotated with named entity mentions. The authors map all hyperlinks in Wikipedia abstracts to entity types using DBpedia [9] attributes. Then, they use neural model trained on DocRED [172] to augment the dataset with false negative (missed) entity type annotations. Results shows that models (e.g. BERT) trained on AnchorNER perform well in cross-domain evaluation. HAnDS [3] is a framework that extends the methods of Ling and Weld [91] by augmenting Wikipedia with internal links using heuristic rules for fine-grained entity typing.

3.7 Conclusion

In this chapter, we presented the task of named entity recognition, a prerequisite for many NLP applications such as dialogue systems and question answering. We give an overview on dataset, features, models and evaluation metrics for NER. Then, we review

5. during the writing of this document

6. Also, they use different evaluation process which make the comparison difficult, thus we leave to future work

7. An equivalent of Freebase

distant supervision methods that aim to automatically generate annotated data for NER. We choose NER as the main application to demonstrate and evaluate the usefulness of distant supervision data either as training data for supervised models or as a resource to learn representation.

In this thesis, we propose a pre-processing pipeline to automatically extract annotations from Wikipedia that is mainly based on coreference. While previous works use Wikipedia as is or with simple matching heuristics, we build on their works by deeply exploiting coreference resolution for entity in Wikipedia. That is, we extend previous works by exploiting special type of *coreference heuristics* that are designed based on the structure of Wikipedia. The main goal is to increase the quantity of labeled examples, while introducing the a reasonable amount of noise. We propose an iterative annotation procedure that follow an easy first strategy (high to low precision matching).

CHAPTER 4

FROM WIKIPEDIA TO WINER AND WIFINE

4.1 Overview

In this part of the thesis, our goal is to augment Wikipedia with as much semantic information as possible by detecting missing hyperlinks. Our approach relies on a heuristic labelling method to automatically generate training data from Wikipedia. We only use the structure of Wikipedia itself, and information from Freebase in order to enrich Wikipedia with missing links. Then, we assign to these links a coarse and fine entity types, leading to 2 corpora: WiNER and WiFiNE respectively. This chapter presents the methodologies we follow to obtain the aforementioned corpora. Content of this chapter¹ was published in:

- Abbas Ghaddar and Philippe Langlais. WikiCoref: An English Coreference-annotated Corpus of Wikipedia Articles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Portorož, Slovenia, 05/2016 2016
- Abbas Ghaddar and Phillippe Langlais. Coreference in Wikipedia: Main concept resolution. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 229–238, 2016
- Abbas Ghaddar and Phillippe Langlais. Winer: A wikipedia annotated corpus for named entity recognition. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 413–422, 2017
- 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

1. the first 2 publication were done during my master [51]

4.2 A Four Stage Approach

Given a Wikipedia page, we attempt to find proper name, nominal and pronominal mentions that refer to an entity (or concept) in Wikipedia. We propose an approach that mainly relies on Wikipedia markup (anchor link, redirect, link-out), and structured data from Freebase (gender, number, aliases, etc.). Our approach follows an easy-first strategy in order to increment Wikipedia from highest to lowest precision annotations. We treat the task as a Coreference Resolution (CR) problem, and define a pipeline of 4 steps:

- (1) Find Main Concept (MC) mentions in each Wikipedia Page.
- (2) Track entities that don't appear as anchored links.
- (3) Resolve proper and nominal coreference mentions of entity links.
- (4) Resolve pronominal coreference using an adapted version of the rule-based coreference model of [129].

An overview of our annotation process is illustrated in Figure 4.1, and subsequent sections will detail the process.

{Chilly Gonzales} (born {Jason Charles Beck}; 20 March 1972) is a [Canadian] musician who resided in [Paris], [France] for several years, and now lives in [Cologne], [Germany]. Though best known for {his} first MC [...], {he} is also a pianist, producer, and songwriter. {The performer} was signed to a three-album deal with Warner Music Canada in 1995, a subsidiary of [Warner Bros. Records] ... While the album's production values were limited [Warner Bros.] simply ...

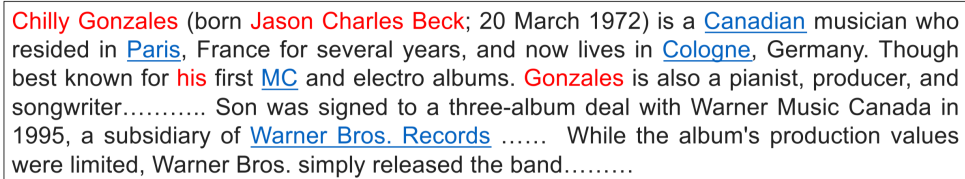
Paris	↪ Europe, France , Napoleon, ...
Cologne	↪ Germany , Alsace, ... OLT
Warner Bros. Records	↪ Warner, Warner Bros. , the label, ...
France	↪ French Republic, the country... CT

Figure 4.1 – Illustration of the process with which we gather annotations into WiNER for the target page https://en.wikipedia.org/wiki/Chilly_Gonzales. Square Bracketed segments are the annotations; curly brackets indicate main concept mentions from Ghaddar and Langlais [55]; while underlined text are anchored texts in the corresponding Wikipedia page. OLT represents the out-link table (which is compiled from the Wikipedia out-link graph structure), and CT represents the coreference table we gathered from the resource.

4.2.1 Main Concept Mention Detection

The first step in our pipeline consists in identifying in a Wikipedia article all the mentions of the concept being described by an article. This part of the annotation process was done during the master thesis [51], thus we give only a general overview of this step.

We refer to this concept as the “main concept” (MC) henceforth. For instance, within the article `Chilly_Gonzales`, the task is to find all proper (e.g. *Gonzales*, *Beck*), nominal (e.g. *the performer*) and pronominal (e.g. *he*) mentions that refer to the MC “Chilly Gonzales”. Due to the specificity of Wikipedia articles (the entire article describe one concept), these mentions can be identified with high accuracy. According to [52] 25% of coreferential mentions in Wikipedia refer to main concepts. We frame the task of MC detection as a binary classification problem, where one has to decide whether a detected mention refers to the MC. We use an SVM classifier [36] that exploits carefully designed features extracted from Wikipedia markup and characteristics, as well as structured data from Freebase.



Chilly Gonzales (born Jason Charles Beck; 20 March 1972) is a Canadian musician who resided in Paris, France for several years, and now lives in Cologne, Germany. Though best known for his first MC and electro albums. Gonzales is also a pianist, producer, and songwriter..... Son was signed to a three-album deal with Warner Music Canada in 1995, a subsidiary of Warner Bros. Records While the album's production values were limited, Warner Bros. simply released the band.....

Figure 4.2 – The first step of our process: main concept mentions detection. Given the article of Chilly Gonzales the goal is to find all nominal and pronominal mentions (red span) that refer to *Chilly Gonzales*.

The classifier was trained on WikiCoref [52], 30 English Wikipedia articles manually coreference-annotated. It comprises 60k tokens annotated with the OntoNotes project guidelines [125]. Each mention is annotated with three attributes: the mention type (named-entity, noun phrase, or pronominal), the coreference type (identity, attributive or copular) and the equivalent Freebase entity if it exists. The resource contains roughly 7 000 non singleton mentions, among which 1 800 refer to the main concept, which is to say that 30 chains out of 1 469 make up for 25% of the mentions annotated. The classifier trained on WikiCoref can detect MC mentions with an an accuracy of 89%.

We generate a special Wikipedia dump² of all the mentions in English Wikipedia (version of April 2013) that our classifier identified as referring to the main concept, along with information we extracted from Wikipedia and Freebase. We build on top of this dump to increment each article with non primary (secondary) entity links.

4.2.2 Secondary Entities Mentions Detection

Within an article, we refer to Wikipedia entities that are different from the main concept as secondary entities. That is, all Wikipedia entries that are present in the article of *Chilly Gonzales* (e.g. *France*, *Warner Bros.* in Figure 4.2) except the main concept entity. We follow a similar path as Raganato et al. [128] in order to identify secondary entities mentions, that is, by using string matching heuristics. We differ from their approach by: (1) attempting to solve nominal and pronominal mentions; (2) incorporating entities from Freebase; (3) using an extended list of coreferent mentions. In addition, we adopt an easy-first strategy in order to select high confident annotation first, which reduces errors.

Because the number of anchored texts in Wikipedia is rather small — less than 3% of the text tokens according to [5] — we propose to leverage the out-link structure of Wikipedia, coupled with the information of all the surface strings used in a Wikipedia article used to express the main concept being described. For the latter, we rely on a the resource that described in Section 4.2.1 that lists, for all the articles in Wikipedia, all the text mentions that are coreferring to the main concept of an article. Our strategy for collecting extra annotations is described in the following:

Following out-links We follow out-links of out-links, and search in the target article (by an exact string match) the titles of the articles reached. Figure 4.3 illustrate this process, in the left part of the figure, we search for the strings *Europe*, *France*, *Napoleon*, as well as other article titles from the out-link list of the article *Paris*.

2. <http://rali.iro.umontreal.ca/rali/?q=en/wikicoref>



Figure 4.3 – Two examples that illustrate the first step of our process: link detection by following link out of link out.

Proper and nominal mentions coreference We consider direct out-links of the target article (*Chilly Gonzales* in our ongoing example). We search the titles of the articles we reach that way. We also search for their coreferences as listed in the main concept resource of [55]. As illustrated in Figure 4.4, we search (exact match) *Warner Bros. Records* and its coreferences (e.g. *Warner*, *Warner Bros.*) in the target article.

[Chilly Gonzales](#) (born [Jason Charles Beck](#); 20 March 1972) is a [Canadian](#) musician who resided in [Paris](#), [France](#) for several years, and now lives in [Cologne](#), [Germany](#). Though best known for [his](#) first [MC](#) and electro albums. [Gonzales](#) is also a pianist, producer, and songwriter..... [Son](#) was signed to a three-album deal with [Warner Music Canada](#) in 1995, a subsidiary of [Warner Bros. Records](#) While the album's production values were limited, [Warner Bros.](#) simply released the band.....

- [Chilly Gonzales](#) → {Gonzales, Jason Charles Beck, the performer}
- [France](#) → {French Republic, Kingdom of France, the country}
- [Warner Bros. Records](#) → {[Warner Bros.](#), Warner, the company}

Figure 4.4 – The second step of our process: proper and nominal coreference mention detection. The text box is our outgoing example, and the bullet list contains coreference mention of entities that we match in the paragraph (e.g. *Warner Bros.*).

Pronominal mentions coreference Last, we adapt the multi-sieve rule-based coreference resolver of [129] to the specificity of Wikipedia in order to find pronominal mentions antecedent referents³. The heuristic rules⁴ link a pronoun to its best antecedent mention based on attributes agreement (same gender, number, entity type, etc ...). We apply the pronominal coreference resolver on each article, then discard all pronouns that do not refer to a Wikipedia entity mention.

4.2.3 Manual evaluation of link augmentation

During this process, some collisions may occur. We solve the issue of overlapping annotations by applying the steps exactly in the order presented above. Our steps have been ordered in such a way that the earlier the step, the more confidence we have in the strings matched at that step. It may also happen that two out-link articles contain the same mention (for instance `Washington_State` and `George_Washington` both contain the mention `Washington`), in which case we annotate this mention with the type of the nearest full name already annotated. Step 1 raises the coverage from less than 3% to 9.5%, step 2 raise it to 11.5 %, while step 3 and 4 increase it to 23% and 30% respectively. Our corpus actually contains many more annotations than existing Wikipedia-based annotated corpora as shown in Table 4.I.

Corpus	# Links	# Entities	# Documents	Link/Doc
Wikipedia (2014)	71.5	2.9	4.3	16.6
Wikilinks [145]	40.3	2.9	10.9	3.7
ClueWeb12 [47]	11240	5.1	1104	10.2
SWE [128]	162.6	4.0	4.3	37.8
Our method	182.7	3.0	3.2	57.0

Table 4.I – Comparison of different link-enriched corpora. Counts in columns Links, Entities and Documents are in millions.

[PER Chilly Gonzales] (born [PER Jason Charles Beck]; 20 March 1972) is a [MISC Canadian] musician who resided in [LOC Paris], [LOC France] for several years, and now lives in [LOC Cologne], [LOC Germany]. Though best known for his first [MISC MC] and electro albums, [PER Gonzales] is also a pianist, producer, and songwriter..... [MISC Son] was signed to a three-album deal with [ORG Warner Music Canada] in 1995, a subsidiary of [ORG Warner Bros. Records] While the album's production values were limited, [ORG Warner Bros.] simply released the band.....

Figure 4.5 – All entity links in our outgoing example are mapped to 4 entity types according to CONLL-2003 annotation scheme.

4.3 WiNER

In general, a Wikipedia article has an equivalent page in Freebase. We associate a category with each mention by a simple strategy, similar to Al-Rfou et al. [5], which consists in mapping Freebase attributes to CONLL-2003 four classes annotation scheme as shown in Figure 4.5. We map `organization/organization`, `location/location` and `people/person notable_type` attributes to ORG, LOC and PER, respectively. If an entry does not belong to any of the previous classes, we tag it as MISC. Figure 4.6 illustrates the mapping of the entity link *Paris* from its Wikipedia -> Freebase page -> `notable_type` attribute to the LOC type.

WiNER contains 3.2M Wikipedia articles, comprising more than 1.3G tokens accounting for 54M sentences, 41M of which contain at least one named-entity annotation. Table 4.II shows the counts of token strings annotated with at least two types. For instance, there are 230k entities that are annotated in WiNER as PER and LOC. It is reassuring that different mentions with the same string are labelled differently. The cells on the diagonal indicate the number of mentions labelled with a given tag.

We generated a total of 106M annotations (an average of 2 entities per sentence). We manually examined a random subset of 100 strings that were annotated differently (in different contexts) and found that 89% of the time, the correct type was identified. For instance in the sentence I didn't want to open up my Rolodex and get everyone to sing for me in the Chilly_Gonzales article, the men-

3. e.g. finding the *it* or *they* that refer to *France* and *Warner Bros.* respectively
 4. last sieve of Raghunathan et al. [129] system, please refer to the article for more details

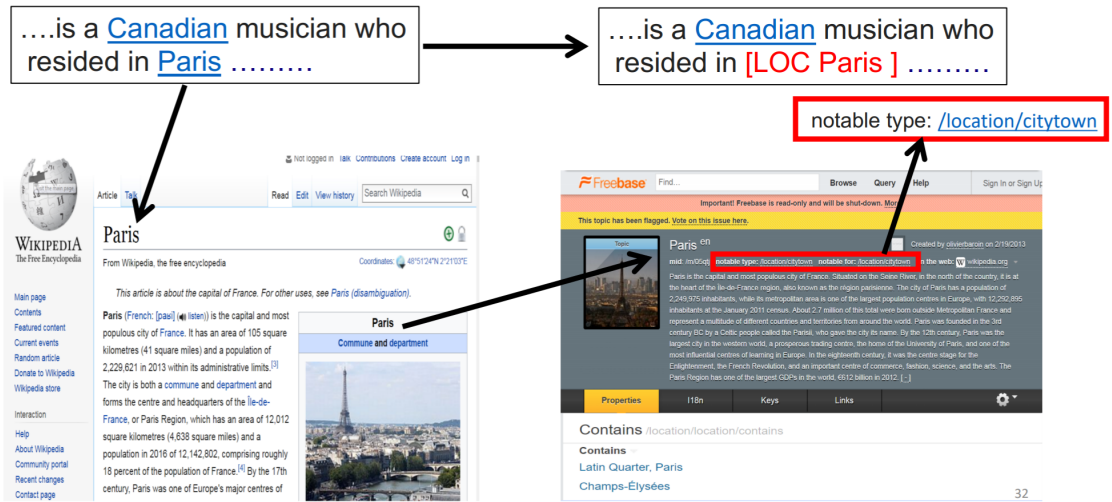


Figure 4.6 – Illustration of mapping entity link (Paris in this example) to named entity annotation through Freebase.

	PER	LOC	ORG	MISC
PER	28M	230k	80k	250k
LOC	-	29M	120k	190k
ORG	-	-	13M	206k
MISC	-	-	-	36M

Table 4.II – Number of times a text string (mention) is labelled with (at least) two types in WiNER. The cells on the diagonal indicate the number of annotations.

tion *Rollodex* was labelled as ORG, while the correct type is MISC. Our process failed to disambiguate the company from its product.

4.4 WiFiNE

We created a second version of the corpus that contains fine-grained entity type mentions. WiFiNE gathers 182.7M mentions: 95.1M proper, 62.4M nominal and 24.2M pronominal ones. Following previous works, we map Freebase *notable_type* attributes of each entity mention detected to a set of fine-grained types. In the last few years, two popular mapping schemes (see Figure 4.8) emerged: FIGER [91] (112 label) and GILLICK [58] (89 label).

They are both organized in a hierarchical structure, where children labels also inherit

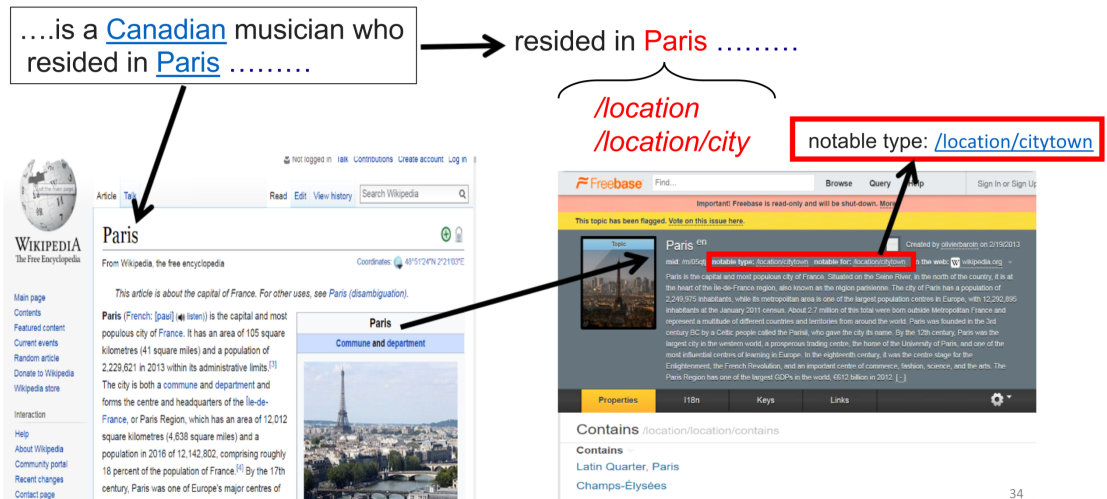


Figure 4.7 – Illustration of mapping entity link (Paris in this example) to hierarchical fine grained entity types through Freebase.

the parent label. FIGER defines a 2-level hierarchy (e.g. */person* and */person/musician*); while GILLICK uses 3 levels of types (e.g. */person* and */person/artist*, */person/artist/musician*).

An entity mention is said to be clean if its labels belong to only a single path (not necessarily a leaf); otherwise, it is noisy. For example, the mentions *France* or *Germany* with labels */location* and */location/country* are considered clean. On the other hand, the entity mention *Chilly Gonzales* annotated with 5 labels (*/person*, */wikipedia/artist*, */person/artist/musician*, */person/artist/actor*, and */person/artist/author*) is considered noisy because only one of the last three types is qualified in a given context.

Most resolved entities have multiple type labels, but not all of them typically apply in a given context. One solution consists in ignoring the issue, and instead relying on the robustness of the model to deal with heterogeneous labels; this approach is adopted by [143, 173]. Another solution involves filtering. In [58, 91], the authors apply hard pruning heuristics:

- **Sibling pruning** Removes sibling types if they came from a single parent type. For instance, a mention labelled as */person/artist/musician* and

person actor architect artist athlete author coach director	doctor engineer monarch musician politician religious_leader soldier terrorist	organization airline company educational_institution fraternity_sorority sports_league sports_team	terrorist_organization government_agency government political_party educational_department military news_agency
location city country county province railway road bridge	body_of_water island mountain glacier astral_body cemetery park	product engine airplane car ship spacecraft train	camera mobile_phone computer software game instrument weapon
building airport dam hospital hotel library power_station restaurant sports_facility theater	time color award educational_degree title law ethnicity language religion god	chemical_thing biological_thing medical_treatment disease symptom drug body_part living_thing animal food	art written_work film newspaper play music event military_conflict attack natural_disaster election sports_event protest terrorist_attack website broadcast_network broadcast_program tv_channel currency stock_exchange algorithm programming_language transit_system transit_line

(a)

PERSON	LOCATION	ORGANIZATION	OTHER	
artist actor author director music education student teacher athlete business coach doctor legal military political figure religious leader title	structure airport government hospital hotel restaurant sports facility theatre geography body of water island mountain transit bridge railway road celestial city country park	company broadcast news education government military music political party sports league sports team stock exchange transit	art broadcast film music stage writing event accident election holiday natural disaster protest sports event violent conflict health malady treatment award body part currency	language programming language living thing animal product camera car computer mobile phone software weapon food heritage internet legal religion scientific sports & leisure supernatural

(b)

Figure 4.8 – (a) FIGER [91] annotation scheme consists of 112 entity types that are stored in a 2 level hierarchical structure (red rectangles indicate parent types). (b) GILLICK [58] defines 3 levels of types, a total of 89 labels (separated by boxes). source [91] and [58]

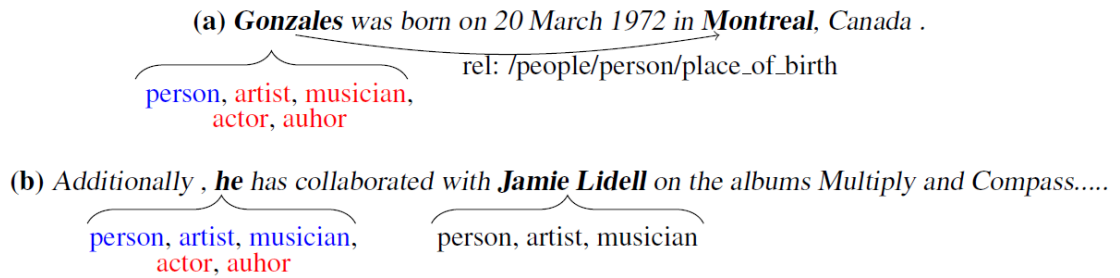


Figure 4.9 – Illustration of the de-noise heuristics rules. Spans in bold are entity mentions, and blue labels are relevant labels, while red are irrelevant ones.

/person/artist/actor would be tagged by /person/artist and /person.

- **Minimum count pruning** All labels that appear once in the document are removed. For example, if multiple entities in a document are labelled as /person/artist/musician and only one of them have /person/artist/actor as an extra label, the latter is considered noisy.

Such heuristics decrease the number of training data by 40-45% according to [58, 136]. Ren et al. [136] propose a distant supervision approach to deal with noisy labelled data. Their method consists in using unambiguous mentions to de-noise mentions with heterogeneous labels that appear in a similar context.

We measured that 23% of mentions in WiFiNE that have two labels or more do not belong to a single path (noisy), and 47% of those have more than 2 noisy labels (e.g. *Gonzales* in Fig. 4.9). We propose to eliminate noisy labels in WiFiNE using rules based on the high coverage of entity mentions, coupled with Freebase triples and the paragraph and section structure of Wikipedia:

1. **Freebase Relation Type:** We label the mention by the type indicated by the relation. A Freebase relation is a concatenation of a series of fragments. The first two fragments of the relation indicate the Freebase type of the subject, and the third fragment indicates the relation type. In example (a) of Fig. 4.9, the triple (arg1: *Chilly Gonzales*; rel: /people/person/place_of_birth; arg2: *Montreal*) found in Freebase indicates that only /person should apply to the

Gonzales mention in this context.

2. **Common Attribute Sharing:** If a non-ambiguous mention (*Jamie Lidell* in example (b)) has a type set which is a subset of another mention with noisy labels (*he*, referent of *Chilly Gonzales*) occurs in the same sentence, we assign to the noisy mention the common labels between both mentions.

We first apply our rules at the sentence level, then at the paragraph and section level. Whenever we de-noise an entity mention in such a way, all its coreferent mentions (in the scope) receive the same type.

Heuristic	Pre	Rec	F1
w/o Rules	31.8	100.0	48.3
Rule-1 only	48.8	87.2	62.3
Rule-2 only	56.4	85.6	68.0
Both Rules	79.2	81.8	80.5
Level of Application			
Sentence	66.5	85.5	73.7
+ Paragraph	72.7	82.6	78.6
+ Section	79.2	81.8	80.5

Table 4.III – De-noising rules evaluation on 1000 hand-labelled mentions following GILLICK type hierarchy.

We assessed the quality of our de-noising rules on 1000 randomly selected noisy mentions. Table 4.III reports precision, recall and F1 scores on the ablation study of the proposed heuristics. We start with an accuracy of 48% when no rule is applied. We measure performance after removing labels identified as noisy by rule one, two and both. Also, we measure the accuracy when the rules are applied at sentence, paragraph and section levels. Results show that our rules greatly improve the annotation quality by roughly 32%. Also, we observe that the first rule is more important than the second, but both rules complement each other. As expected, applying the rules at paragraph and section levels further improve the performance.

We identify two sources of errors: (1) pruning heuristics do not apply to 11% of mentions; (2) our rules failed to pick up the correct label in 9% of the cases. Ex-

Sentence	Labels
In Kent v. Dulles , 357 U.S. 116 (1958) , the Court held that the federal government ...	/other /other/event
The Cangrejaj River or Río Cangrejaj is a river that drains several mountain tributaries ...	/location /location/geography /location/geography/body_of_water
...editions of Millionaire to be aired between 7:00 and 7:30 pm	/other /other/art /other/art/broadcast
Mies Bouwman stopped her regular work after falling sick but has occasionally	/person /person/artist
... to imprisoned Christians and niece of the Emperor Gallienus , found Anthimus in prison .	/person /person/political_figure
... of vinyl siding which does not weather as wood does .	/other /other/product
The firm was the first state-owned rail vehicle in Argentina...	/organization /organization/company
The 1 – 2 ton was a sailing event on the Sailing at the 1900 Summer Olympics program in Meulan	/other /other/event /other/event/sports_event
He took part in the White Council after Sauron 's return....	/person /person/artist /person/artist/actor
Clove is Syzygium aromaticum and belongs to division of Magnoliophyta in the kingdom Plantae .	/other /other/living_thing
Pepsi * also created a fellowship at Harvard University which enable students from...	/other /other/food
... Viitorul Homocea * , Siretul Suraia and Trotusul Ruginesii deducted 3 points .	/location

Table 4.IV – Random selection of annotations from WiFINE following GILICK type hierarchy. Faulty annotations are marked with a star.

ample (a) of Figure 4.10 illustrates such a mistake where *Gonzales* is labelled as *musician* rather than *author* because *Feist* is considered as *musician* in this context. In example (b), *Gonzales* is wrongly labelled as *person* even though the relation /people/person/nationality exists between both entities but the sentence does not state it.

- a) *[Gonzales]_{musician*}* returned as a contributor on *[Feist]_{musician}*'s 2007 album...
- b) *[Gonzales]_{person*}* said in an interview: My experiences in *[Canada]_{country}* had been disappointing

Figure 4.10 – Examples of errors in our de-noising rules. Faulty annotations are marked with a star.

Table 4.IV illustrates a randomly-picked selection of mentions annotated in WiFiNE, along with their type according to the GILLICK scheme. The last two examples illustrate noisy annotations. In the first example⁵, our process failed to distinguish between the company and its product. The second example is a mention detection error, we couldn't recognize *Viitorul Homocea* as an entity, because this soccer team does not have a page in Wikipedia or Freebase.

4.4.1 Corpus Statistics

WiFiNE is built from 3.2M Wikipedia articles, comprising more than 1.3G tokens accounting for 54M sentences, 41M of which contain at least one entity mention. Overall, it gathers 182.7M mentions: 95.1M proper, 62.4M nominal and 24.2M pronominal ones. Table 4.V summarizes the mention statistics and label distribution over the number of levels of FIGER and GILLICK type hierarchies.

First, we note that the total number of mentions in FIGER and GILLICK is less than the total number of entity mentions. This is because: (a) we remove noisy mentions that our rules failed to disambiguate (11%), (b) some mentions cannot be mapped to either

5. second last line in table 4.IV

	FIGER	GILLICK
Total mentions	159.4	111.1
Proper mentions	82.5 (52%)	64.8 (58%)
Nominal mentions	55.9 (35%)	29.8 (27%)
Pronominal mentions	21.0 (13%)	16.5 (15%)
Total Labels	243.2	230.9
Level 1	153.8 (63%)	111.1 (48%)
Level 2	89.5 (37%)	90.0 (39%)
Level 3	-	29.8 (13%)

Table 4.V – Mention statistics and label distribution (in millions and percentages) over the number of levels of FIGER and GILLICK type hierarchy.

schemes (e.g. fictional characters). Second, we note that FIGER mentions outnumber those of GILLICK, simply because their scheme covers more types (112 vs 89). Following the GILLICK scheme, each mention has 2 types on average, where 39% of them are of level 2, and 13% are of level 3. The distribution of level 2 and 3 labels in WiFiNE exceed its equivalent in the ONTONOTES [58] dataset (29% and 3% respectively).

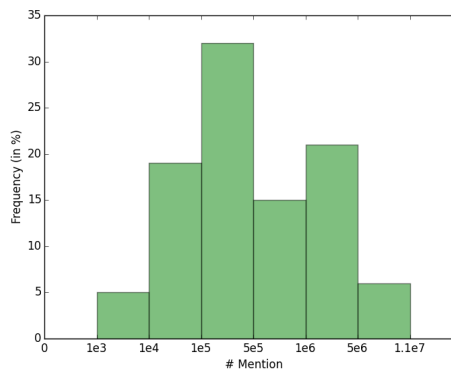


Figure 4.11 – Distribution of entity type labels according to the FIGER type hierarchy.

Figure 4.11 illustrates the percentage of types that receive a given number of mentions in WiFiNE. It shows that the majority of types have more than 100k mentions and roughly 25% (like `city`, `company`, `date`) exceeds 1M mentions. Also, we observe that 5% of the types have less than 10k mentions (e.g. `/event/terrorist_attack`),

and none of them has less than 1k mentions⁶.

4.5 Conclusion

We revisited the task of using Wikipedia for generating annotated data suitable for entity detection and typing. We significantly extended the number of annotations of non anchored strings in Wikipedia, thanks to coreference information and an analysis of the link structure. We applied our approach to a dump of English Wikipedia from 2013, leading a corpus of enriched links which surpasses other similar corpora in term of density. Then, we map entity links to coarse and fine entity type annotations leading to WiNER and WiFiNE respectively. In the next Chapter, we will evaluate annotation quality extrinsically by using the corpus as training data for: named-entity recognition and fine-grained entity typing tasks.

6. A similar distribution is obtained with GILLICK type hierarchy.

CHAPTER 5

DISTANT SUPERVISION DATA FOR TRAINING

5.1 Overview

In this chapter, we focus on usefulness of WiNER and WiFiNE as training data for entity typing tasks, the traditional approach for evaluating distant supervision. We perform extensive experiments on NER and fine-grained entity typing with off-the-shelf toolkits. This allows us to compare WiNER and WiFiNE with distant supervision training data. In order to ensure fair and meaningful comparison between datasets, we tried as possible to train all systems under the same conditions (such as parameters, resource, training time and size). The chapter is divided into 2 parts: experiments on NER with WiNER in Section 5.2, and experiments on fine-grained entity typing with WiFiNE in Section 5.3. Content of this Chapter was published in:

- Abbas Ghaddar and Phillippe Langlais. Winer: A wikipedia annotated corpus for named entity recognition. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 413–422, 2017
- 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

5.2 Experiments on WiNER

We explore WiNER by using it as training material to improve NER models behavior in three different scenarios:

1. WiNER compared with distant supervision corpus of previous work in Section 5.2.4.
2. WiNER to improve cross-domain (generalization) performances in Section 5.2.5.
3. Scale to full WiNER with a simple and fast Named Entity classifier in Sec-

tion 5.2.6.

Just before getting into our experiments, we introduce gold-standard datasets used in this study (section 5.2.1). We provide a brief description of various NER systems used in this work (section 5.2.2). Section 5.2.3 describes the evaluation metrics.

5.2.1 Data Sets

We used a number of datasets in our experiments. For CONLL, MUC and ONTONOTES, that are often used to benchmark NER, we used the test sets distributed in official splits. For the other test sets, that are typically smaller, we used the full dataset as a test material.

CONLL the CONLL-2003 NER Shared Task dataset [154] is a well known collection of Reuters newswire articles that contains a large portion of sports news. It is annotated with four entity types (PER, LOC, ORG and MISC).

MUC the MUC-6 [28] dataset consists of newswire articles from the Wall Street Journal annotated with PER, LOC, ORG, as well as a number of temporal and numerical entities that we excluded from our evaluation for the sake of homogeneity.

ONTO the OntoNotes 5.0 dataset [123] includes texts from five different text genres: broadcast conversation (200k), broadcast news (200k), magazine (120k), newswire (625k), and web data (300k). This dataset is annotated with 18 fine grained NE categories. Following [111], we applied the procedure for mapping annotations to the CONLL tag set. We used the CONLL 2012 [124] standard test set for evaluation.

WGOLD WikiGold [13] is a set of Wikipedia articles (40k tokens) manually annotated with CONLL-2003 NE classes. The articles were randomly selected from a 2008 English dump and cover a number of topics.

WEB Ratinov and Roth [132] annotated 20 web pages (8k tokens) on different topics with the CONLL-2003 tag set.

TWEET Ritter et al. [138] annotated 2400 tweets (comprising 34k tokens) with 10 named-entity classes, which we mapped to the CONLL-2003 NE classes.

5.2.2 Reference Systems

We chose two feature-based models: the `StanfordNER` [50] CRF classifier, and the perceptron-based `Illinois NE Tagger` [132]. Those systems have been shown to yield good performance overall. Both systems use handcrafted features; the latter includes gazetteer features as well.

We also deployed two neural network systems: the one of [33], as implemented by Attardi [8], and the `LSTM-CRF` system of Lample et al. [78]. Both systems capitalize on representations learnt from large quantities of unlabeled text.¹ We use the default configuration for each system.

5.2.3 Metrics

Since we use many test sets in this work, we are confronted with a number of inconsistencies. One is the definition of the `MISC` class, which differs from a dataset to another, in addition to not being annotated in `MUC`. This led us to report token-level F1 score for 3 classes only (`LOC`, `ORG` and `PER`). We computed this metric with the `conlleval` script.²

We further report OD_{F1} , a score that measures how well a named-entity recognizer performs on out-domain material. We compute it by randomly sampling 500 sentences³ for each out-domain test set, on which we measure the token-level F1. Sampling the same number of sentences per test set allows weight each corpus equally. This process is repeated 10 times, and we report the average over those 10 folds. On average, the newly assembled test set contains 50k tokens and roughly 3.5k entity mentions. We excluded the `CoNLL-2003` test set from the computation since this corpus is in-domain⁴ (see section 5.2.5).

1. We use the pre-trained representations.

2. <http://www.cnts.ua.ac.be/conll2000/chunking/conlleval.txt>

3. The smallest test set has 617 sentences

4. Figures including this test set do not change drastically from what we observe hereafter.

5.2.4 Comparing with other Wikipedia-based Corpora

We compare WiNER to existing Wikipedia-based annotated corpora. Nothman et al. [112] released two versions of their corpus, WP2 and WP3, each containing 3.5 million tokens. Both versions enrich the annotations deduced from anchored texts in Wikipedia by identifying coreferences among NE mentions. They differ by the rules used to conduct coreference resolution. We randomly generated 10 equally-sized subsets of WiNER (of 3.5 million tokens each).

On each subset, we trained the `Illinois` NER tagger and compared the performances obtained on the `CoNLL` test set by the resulting models, compared to those trained on WP2 and WP3. Phrase-level F1 score are reported in Table 5.I. We also report the results published in [5] with the Polyglot corpus, which is unfortunately not available.

	with MISC	w/o MISC
WP2	68.2	72.8
WP3	68.3	72.9
Polyglot	-	71.3
WiNER	71.2 [70.3,71.6]	74.5 [73.4,75.2]

Table 5.I – Performance of the `Illinois` toolkit on `CoNLL`, as a function of the Wikipedia-based training material used. The figures on the last line are averaged over the 10 subsets of WiNER we randomly sampled. Bracketed figures indicate the minimum and maximum values.

Using WiNER as a source of annotations systematically leads to better performance, which validates the approach we described in Chapter 3. Note that in order to generate WP2 and WP3, the authors applied filtering rules that are responsible for the loss of 60% of the annotations. Al-Rfou et al. [5] also perform sentence selection. We have no such heuristics here, but we still observe a competitive performance. This is a satisfactory result considering that WiNER is much larger.

5.2.5 Cross-domain Evaluation

In this experiment, we conduct a cross-domain evaluation of the reference systems described in Section 5.2.2 on the six different test sets presented in Section 5.2.1. Following a common trend in the field, we evaluate the performance of those systems when they are trained on the CONLL material. We also consider systems trained on CONLL plus a subset of WiNER. We report results obtained with a subset of 3 million tokens randomly chosen, as well as a variant where we use as much as possible of the training material available in WiNER. Larger datasets were created by randomly appending material to smaller ones. Datasets were chosen once (no cross-validation, as that would have required too much time for some models). Moreover, for the comparison to be meaningful, each model was trained on the same 3M dataset. The results are reported in Table 5.II.

	CoNLL	ONTONOTES	MUC	TWEET	WEBPAGES	WIKIGOLD	OD _{F1}
CRF							
CoNLL	91.6	70.2	80.3	38.7	61.9	68.4	67.0
+WiNER(3M)	-	-	-	-	-	-	-
+WiNER(1M)	89.3 (-2.4)	71.8 (+1.7)	78.6 (-1.8)	49.2 (+10.5)	63.0 (+1.1)	69.1 (+0.8)	69.2(+2.2)
Illinois							
CoNLL	92.6	71.9	84.1	44.9	57.0	71.4	68.3
+WiNER(3M)	85.5 (-6.9)	71.4 (-0.5)	76.2 (-7.9)	51.1 (+6.2)	65.5 (+8.5)	71.8 (+0.4)	69.5(+1.2)
+WiNER(30M)	82.0 (-10.6)	71.6 (-0.3)	75.6 (-8.5)	52.2 (+7.3)	63.3 (+6.3)	71.6 (+0.3)	69.0(+0.7)
Senna							
CoNLL	90.3	68.8	73.2	36.7	58.6	70.0	64.3
+WiNER(3M)	86.6 (-3.7)	70.1 (+1.3)	73.9 (+0.7)	43.2 (+6.4)	62.6 (+4.0)	69.9 (-0.1)	67.0(+2.7)
+WiNER(7M)	86.8 (-3.5)	70.0 (+1.2)	72.9 (-0.4)	44.8 (+8.1)	61.5 (+2.9)	69.3 (-0.7)	66.2(+1.9)
LSTM-CRF							
CoNLL	92.3	71.3	76.6	36.7	57.4	68.0	65.0
+WiNER(3M)	91.5 (-0.8)	74.7 (+3.4)	84.7 (+8.1)	48.1 (+11.4)	62.7 (+5.2)	73.2 (+5.2)	72.0 (+7.0)
+WiNER(5M)	91.1 (-1.2)	76.6 (+5.3)	84.0 (+7.4)	48.4 (+11.7)	64.4 (+7.0)	74.3 (+6.4)	73.0 (+8.0)

Table 5.II – Cross-domain evaluation of NER systems trained on different mixes of CoNLL and WiNER. Figures are token-level F1 score on 3 classes, while figures in parentheses indicate absolute gains over the configuration using only the CoNLL training material.

First, we observe the best overall performance with the LSTM-CRF system (73% OD_{F1}), the second best system being a variant of the Illinois system (69.5% OD_{F1}). We also observe that the former system is the one that benefits the most from WiNER (an absolute gain of 8% in OD_{F1}). This may be attributed to the fact that this model can

explore the context on both sides of a word with (at least in theory) no limit on the context size considered. Still, it is outperformed by the `Illinois` system on the `WEBPAGES` and the `TWEET` test sets. Arguably, those two test sets have a NE distribution which differs greatly from the training material.

Second, on the `CONLL` setting, our results are satisfyingly similar to those reported in [132] and [78]. The former reports 91.06 phrasal-level F1 score on 4 classes, while our score is 90.8. The latter reports an F1 score of 90.94 while we have 90.76. The best results reported so far on the `CONLL` setting are those of [30] with a BiLSTM-CNN model, and a phrasal-level F1 score of 91.62 on 4 classes. So while the models we tested are slightly behind on `CONLL`, they definitely are competitive. For other tasks, the comparison with other studies is difficult since the performance is typically reported with the full tagset.

Third, the best performances are obtained by configurations that use `WiNER`, with the exception of `CONLL`. That this does not carry over to `CONLL` confirms the observations made by several authors [5, 50]. These authors highlight the specificity of `CONLL`'s annotation guidelines as well as the very nature of the annotated text, where sport teams are overrepresented. These teams add to the confusion because they are often referred to with a city name. We observe that, on `CONLL`, the `LSTM-CRF` model is the one that registers the lowest drop in performance. The drop is also modest for the `CRF` model. The `WiNER`'s impact is particularly observable on `TWEET` (an absolute gain of 8.8 points) and `WEBPAGES` (a gain of 5.5), again two very different test sets. This suggests that `WiNER` helps models to generalize.

Last, we observe that systems differ in their ability to exploit large training sets. For the two feature-based models we tested, the bottleneck is memory. We did train models with less features, but with a significantly lower performance. With the `CRF` model, we could only digest a subset of `WiNER` of 1 million tokens, while `Illinois` could handle 30 times more. As far as neural network systems are concerned, the issue is training time. On the computer we used for this work — a Linux cluster equipped with a GPU — training `Senna` and `LSTM-CRF` required over a month each for 7 and 5 millions `WiNER` tokens respectively. This prevents us from measuring the benefit of the

complete WiNER resource.

5.2.6 Scaling up to WiNER

Because we were not able to employ the full WiNER corpus with the NER systems mentioned above, we resorted to a simple method to leverage all the annotations available in the corpus. It consists in decoupling the segmentation of NEs in a sentence — we leave this to a reference NER system — from their labelling, for which we train a local classifier based on contextual features computed from WiNER. Decoupling the two decision processes is not exactly satisfying, but allows us to scale very efficiently to the full size of WiNER, our main motivation here.

5.2.6.1 Contextual representations

Our classifier exploits a small number of features computed from two representations of WiNER. In one of them, each named-entity is bounded by a beginning and end token tags — both encoding its type — as illustrated on line MIX of Figure 5.1. In the second representation, the words of the named entity are replaced by their type, as illustrated on line CONT. The former representation encodes information from both the context and the words of the segment we wish to label while the second one only encodes the context of a segment.

WiNER [Gonzales]_{PER} will be featured on [Daft Punk]_{MISC} .

MIX ⟨B-PER⟩ Gonzales ⟨L-PER⟩ will be featured on ⟨B-MISC⟩ Daft Punk ⟨L-MISC⟩

CONT ⟨PER⟩ will be featured on ⟨MISC⟩ .

Figure 5.1 – Two representations of WiNER’s annotation used for feature extraction.

With each representation, we train a 6-gram backoff language model using `kenLM` [62]. For the MIX one, we also train word embeddings of dimension 50 using `Glove` [118].⁵ Thus, we have the embeddings of plain words, as well as those of token tags. The language and embedding models are used to provide features to our classifier.

5. We used a window size of 5 in this work.

5.2.6.2 Features

Given a sentence and its hypothesized segmentation into named-entities (as provided by another NER system), we compute using the Viterbi algorithm the sequence of token tags that leads to the smallest perplexity according to each language model. Given this sequence, we modify the tagging of each segment in turn, leading to a total of 4 perplexity values per segment and per language model. We normalize those perplexity values so as to interpret them as probabilities. Table 5.III shows the probability given by both language models to the segment *Gonzales* of the sentence of our running example. We observe that both models agree that the segment should be labelled PER. We also generate features thanks to the embedding model.

This time, however, this is done without considering the context: we represent a segment as the sum of the representation of its words. We then compute the cosine similarity between this segment representation and that of each of the 4 possible tag pairs (the sum of the representation of the begin and end tags); leading to 4 similarity scores per segment. Those similarities are reported on line EMB in Table 5.III.

	LOC	MISC	ORG	PER
CONT	0.11	0.35	0.06	0.48
MIX	0.26	0.19	0.18	0.37
EMB	0.39	0.23	0.258	0.46

Table 5.III – Features for the segment *Gonzales* in the sentence *Gonzales will be featured on Daft Punk*.

To these 4 scores provided by each model, we add 16 binary features that encode the rank of each token tag according to one model (does $\langle \text{tag} \rangle$ has rank $\langle i \rangle$?). We also compute the score difference given by a model to any two possible tag pairs, leading to 6 more scores. Since we have 3 models, we end up with 78 features.

5.2.6.3 Training

We use `scikit-learn` [116] to train a Random Forest classifier⁶ on the 29k mentions of the CONLL training data. We adopted this training material to ensure a fair comparison with other systems that are typically trained on this dataset. Another possibility would be to split WiNER into two parts, one for computing features, and the other for training the classifier. We leave this investigation as future work. Because of the small feature set we have, training such a classifier is very fast.

5.2.6.4 Results

We measure the usefulness of the complete WiNER resource by varying the size of the training material of both language models and word embeddings, from 5M tokens (the maximum size the LSTM-CRF mode could process) to the full WiNER resource size.

	CO	ON	MU	TW	WE	WG	OD_{F1}
5M	84.3	72.0	78.7	39.8	61.9	70.2	68.1
50M	86.8	75.6	82.3	44.9	64.7	73.8	71.7
500M	88.9	76.2	84.8	45.8	66.6	75.5	74.1
All	90.5	76.9	85.9	46.6	65.3	77.0	74.7

Table 5.IV – Influence of the portion of WiNER used in our 2-stage approach for the CONLL test set, using the segmentation produced by LSTM-CRF+WiNER(5M). These results have to be contrasted with the last line of Table 5.II.

To this end, we provide the performance of our 2-stage approach on CONLL, using the segmentation output by LSTM-CRF+WiNER(5M). Results are reported in Table 5.IV. As expected, we observe that computing features on the same WiNER(5M) dataset exploited by LSTM-CRF leads to a notable loss overall (OD_{F1} of 68.1 versus 73.0), while still outperforming LSTM-CRF trained on CONLL only (OD_{F1} of 65.0). More interestingly, we observe that for all test sets, using more of WiNER leads to

6. We tried other algorithms provided by the platform with less success.

better performance, even if a plateau effect emerges. Our approach does improve systematically across all test sets by considering 100 times more WiNER data than what LSTM-CRF can handle in our case. Using all of WiNER leads to an OD_{F1} score of 74.7, an increase of 1.7 absolute points over LSTM-CRF+WiNER(5M).

	CRF	Illinois	Senna	LSTM-CRF
CONLL	73.6 (+6.6)	74.4 (+6.1)	70.1 (+5.8)	69.7 (+4.7)
+3M	-	74.2 (+4.7)	70.8 (+3.8)	74.8 (+2.8)
+max	73.0 (+2.8)	74.3 (+4.3)	72.0 (+5.8)	74.7 (+1.7)

Table 5.V – OD_{F1} score of native configurations, and of our 2-stage approach (RF) which exploits the full WiNER corpus. Figures in parentheses indicate absolute gains over the native configuration.

Table 5.V reports the improvements in OD_{F1} of our 2-stage approach (RF), which uses all of the WiNER material and the segmentation produced by several native systems. Applying our 2-stage approach systematically improves the performance of the native configuration. Gains are larger for native configurations that cannot exploit a large quantity of WiNER. We also observe that the 2-stage approach delivers roughly the same level of performance ($OD_{F1} \simeq 74$) while using the segmentation produced by the `Illinois` and the `LSTM-CRF` systems.

5.2.6.5 Error Analysis

Table 5.VI indicates the number of disagreements between the LSTM-CRF+WiNER(5M) system (columns) and the 2-stage approach (rows). The table also shows the percentage of times the latter system was correct. For instance, the bottom left cell indicates that, on 38 distinct occasions, the classifier changed the tag PER proposed by the native system to ORG and that is was right in 85% of these occasions. We exclude errors made by both systems, which explains the low counts observed (1.7% is the absolute difference between the two approaches).

We observe that in most cases the classifier makes the right decision when an entity tag is changed from PER to either LOC or ORG (86% and 85% respectively). Most often,

	PER	LOC	ORG
PER	-	50% [12]	25% [12]
LOC	86% [20]	-	21% [28]
ORG	85% [38]	81% [19]	-

Table 5.VI – Percentage of correctness of the 2-stage system (rows) when tagging a named-entity differently than the LSTM-CRF+WiNER(5M) (columns). Bracketed figures indicate the average number of differences over the out-domain test sets.

re-classified entities are ambiguous ones. Our approach chooses correctly mostly by examining the context of the mention. For instance, the entity *Olin* in example (a) of Figure 5.2 is commonly known as a last name. It was correctly re-classified as ORG thanks to its surrounding context. Replacing *its* by *his* in the sentence makes the classifier tag the entity as PER. Similarly, the entity *Piedmont* in example (b) was re-classified as ORG, although it is mostly used as the region name (even in Wikipedia), thanks to the context-based CONT and MIX features that identify the entity as ORG (0.61 and 0.63 respectively).

- (a) ... would give [Olin]_{PER→ORG} access to its production processes ...
- (b) Wall Street traders said [Piedmont]_{LOC→ORG} shares fell partly ...
- (c) ★ ... performed as a tenor at New York City 's [Carnegie Hall]_{ORG→LOC}.

Figure 5.2 – Example of entities re-classified by our 2-stage approach.

Misclassification errors do occur, especially when the native system tagged an entity as ORG. In such cases, the classifier is often misled by a strong signal emerging from one family of features. For instance, in example (c) of Figure 5.2, both MIX — $p(\text{ORG}) = 0.39$ vs. $p(\text{LOC}) = 0.33$ — and EMB — $p(\text{ORG}) = 0.39$ vs. $p(\text{LOC}) = 0.38$ — features are suggesting that the entity should be tagged as LOC, but the CONT signal — $p(\text{LOC}) = 0.63$ vs. $p(\text{ORG}) = 0.1$ — strongly impacts the final decision. This was to be expected considering the simplicity of our classifier, and leaves room for further improvements.

5.3 Experiments on WiFiNE

In order to test the WiFiNE, we pick up the state of the art model (at that time) and re-trained on WiFiNE. We compare the results with that of models that are trained on distant or weak supervision data (including Wikipedia). Experiments on WiFiNE as training data are limited comparing with that on WiNER because fine-grained entity typing is less popular than NER in the community (e.g. few labeled data to support the task). However, in the next chapters we extensively explore WiFiNE as a resource for classic and contextualized word representation learning. The next two sections describe the experimental protocol, while the subsequent ones state and analyze the results we obtain.

5.3.1 Reference System

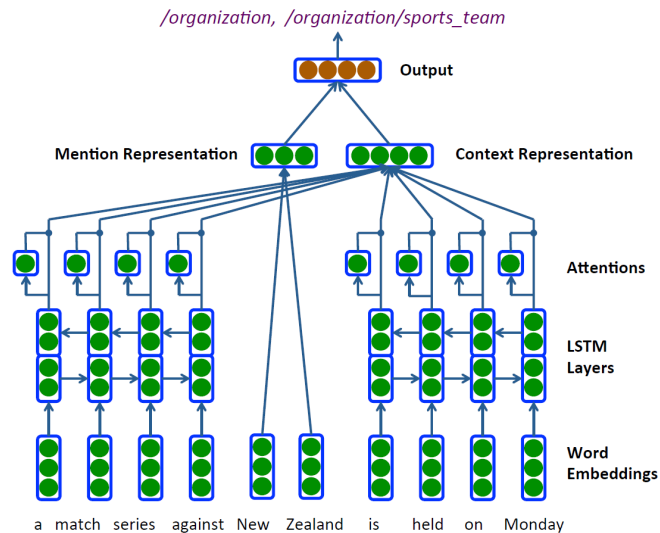


Figure 5.3 – An illustration of the attentive encoder neural model of Shimaoka et al. [143] predicting fine-grained semantic types for the mention “New Zealand” in the expression “a match series against New Zealand is held on Monday”. source [143]

In all experiments, we deploy the off the shelf neural network model of [143]. Given a mention in its context, the model uses three representations in order to associate the mention with the correct types.

- **Mention representation:** the average of the mention words embedding.
- **Context representation:** a Bi-LSTM model is applied on the left and right context of the mention, then an attention layer is placed on top of the model.
- **Feature Representation:** Learning of the representations of hand-crafted features.

We trained the tagger on various subsets of WiFiNE as described in the next section. We use the default configuration of the tagger, except the batch size which we set to 100 rather than 1000 and the learning rate that we changed from 0.001 to 0.0005⁷

5.3.2 Datasets and Evaluation Metrics

We evaluate the model on two manually annotated benchmark: FIGER (GOLD) [91] and ONTONOTES [58]. The first consist of 18 news reports annotated following FIGER scheme, while the second are 77 documents from the OntoNotes 5.0 [163] test set annotated according to the GILLICK scheme. Following previous works, we used Strict, loose Macro-averaged, and loose Micro-averaged F1 scores as metrics for evaluation. Strict measures exact match, while losses metrics measure macro/micro partial matches between gold and system labels. Macro is the average of F1 scores on all types, while Micro is the harmonic mean. Table 5.VII and 5.IX compared the performance obtained by the resulting models with those of previous works on FIGER (GOLD) and ONTONOTES test set respectively. We perform an ablation test on our 4-step process of Section 4.2.2 by training the model on 7 variants of WiFiNE:

- **Line 1-3:** hyperlinks + proper name coreference mentions (step 1 and 2 of Section 4.2.2)
- **Line 4:** hyperlinks + proper name + nominal coreference mentions (step 1-3 of Section 4.2.2).
- **Line 5:** hyperlinks + proper name + pronominal coreference mentions (step 1, 2 and 4 of Section 4.2.2).
- **Line 6-7:** hyperlinks + proper name + nominal + pronominal coreference mentions (all steps).

7. We observed better results on the held-out development set.

The goal is to validate if proper name, nominal and pronominal coreference mentions are necessary for fine-grained entity typing performance. For each variant, we report the average score on 5 randomly generated subsets. To be comparable with previous works, we used training material up to 4 million mentions, and leave experiments on the usefulness of the full WiFiNE for future work.

5.3.3 Results on FIGER (GOLD)

Previous works trained their models on 2.6 million mentions obtained by mapping hyperlinks in Wikipedia articles to Freebase⁸.

Models				Strict	Macro	Micro
FIGER [91]				52.30	69.90	69.30
FIGER+PLE [136]				59.90	76.30	74.90
Attentive [143]				59.68	78.97	75.36
Abhishek et al. [2]				65.80	81.20	77.40
Proper Nominal Pronominal				This work		
(1)	1	0	0	61.99	76.20	75.12
(2)	2	0	0	63.77	77.56	76.25
(3)	3	0	0	63.41	78.03	76.32
(4)	1	1	0	64.83	79.26	77.36
(5)	1	0	1	63.06	79.00	76.77
(6)	1	1	1	65.19	79.59	77.55
(7)	2	1	1	66.07	79.94	78.21

Table 5.VII – Results of the reference system trained on various subsets of WiFiNE, compared to other published results on the FIGER (GOLD) test set. Training data (in millions) include: proper name; nominal and pronominal mentions.

Our model trained on 4M mention (line 7) outperforms the initial model of [143] by 6.2, 1.0 and 2.9 on strict, micro, macro F1 scores, and the state-of-the-art of [2] by 0.3 and 0.9 strict and macro F1 scores. First, we observe that using hyperlinks and proper name mentions (line 3) for training improves the performance of the original model of [143] that uses data driven from hyperlinks only. Second, we notice that models trained on a mix of proper name and nominal (line 4) or pronominal (line 5) coreference

8. The dataset is distributed by [136]

mentions outperform the model trained on proper name mentions (line 2) solely. Third, we observe that the combination of 3 mention types (line 6-7) is required in order to outperform the state-of-the-art, which validate our 4-step method of Section 4.2.2.

Label type	FIGER (GOLD)	WiFiNE
/person	31.5%	16.6%
/organization	16.9%	7.7%
/location	13.2%	13.6%
/location/city	5.0%	4.3%
/organization/sports_team	4%	1.0%

Table 5.VIII – Comparison of the distribution of the top 5 types present in FIGER (GOLD) test set to that of WiFiNE.

Table 5.VIII shows the 5 most frequent types the FIGER (GOLD) test set compared to those in WiFiNE. FIGER (GOLD) is a small dataset, it contains only 523 mentions annotated with 41 different labels. We observe that the type distribution in this dataset follows a zipfian curve, while the distribution of types in WiFiNE is similar to a normal distribution (Figure 4.11). Figure 5.4 illustrates some errors committed on FIGER (GOLD) dataset. Error mostly occur on mentions with labels that don't belong to a single path (example a), and on ambiguous mentions (example b).

(a) ... bring food for the employees at [Safeway] ...

Gold: /location /location/city

/organization /organization/company

Pred: /organization /organization/company

(b) *With the huge popularity of [EyeFi] cards ...*

Gold: /product

Pred: /organization

Figure 5.4 – Examples of mentions erroneously classified in FIGER (GOLD) dataset.

5.3.4 Results on OntoNotes

Ren et al. [136], Shimaoka et al. [143] and Abhishek et al. [2] trained their models

on newswire documents present in OntoNotes [163], where entity mentions were automatically identified and linked to Freebase using DB-pedia Spotlight [38]. On the other hand, Gillick et al. [58] and Yogatama et al. [173] used an entity linker to automatically annotate 113k news documents. Results on the ONTONOTES dataset validate the observation we obtained on FIGER (GOLD). Models trained on proper names in addition to nominal (line 4 in Table 5.IX) or pronominal (line 5) coreference mentions is better than only training on proper names (line 2). In addition, training on the combination of all coreference mentions (line 6-7) systematically improves performances.

Models				Strict	Macro	Micro
Gillick et al. [58]				N/A	N/A	70.0
K-WASABIE [173]				N/A	N/A	72.98
FIGER+PLE [136]				57.20	71.50	66.10
Attentive [143]				51.74	70.98	64.91
Abhishek et al. [2]				52.20	68.50	63.30
Proper Nominal Pronominal				This work		
(1)	1	0	0	55.25	68.21	61.49
(2)	2	0	0	57.05	71.96	66.03
(3)	3	0	0	57.47	72.87	66.97
(4)	1	1	0	57.17	73.07	67.30
(5)	1	0	1	57.50	73.08	67.35
(6)	1	1	1	57.80	73.60	67.82
(7)	2	1	1	58.05	73.72	67.97

Table 5.IX – Results of the reference system trained on various subsets of WiFiNE, compared to other published results on the ONTONOTES test set. Training data (in millions) includes: proper name; nominal and pronominal mentions.

We outperform best results reported by previous works on strict, macro F1 scores by 0.9 and 2.3 respectively. On the other hand, we underperform [58] and [173] and by 3 and 5 point on the micro metric respectively. In [58, 173], the authors do not report results on strict and macro metrics and neither their models nor their training data are available. Consequently, we couldn’t identify the cause of the gap on the micro metric, but we report some improvement over [143] model on the loose metrics. A potential reason for this gap is that the text genre of their training data and that of ONTONOTES is

the same (newswire). Our models were trained on randomly picked Wikipedia sentences (out of domain). Also, we note that in order to generate their corpus, [58, 173] applied filtering rules that are responsible for the loss of 45% of the mentions. We have no such heuristic here, but we still observe competitive performances.

Label type	Onto Test	WiFiNE
/other	44.0%	20.0 %
/organization	10.5%	6.3 %
/person	8.4%	17.6%
/organization/company	7.7%	2.3%
/location	7.6%	11.8%

Table 5.X – Comparison of the distribution of the top 5 types present in ONTONOTES test set to that of WiFiNE.

trouble
addition
personal reasons
some complications
additional evidence
diplomatic relations
a modest pretax gain
the active role taken
in the affairs of United
quotas on various economic indicators
the invitation of the Foreign Affairs Institute
amounts related to areas where deposits are received

Table 5.XI – Examples of non-entity mentions annotated as /other in the of OntoNotes test set.

Table 5.X shows the 5 most frequent types in the ONTONOTES dataset and in WiFiNE. Although ONTONOTES is much larger the FIGER (GOLD)⁹, we still observe that the distribution of types in this dataset is zipfian. We also note that the type /other is over-represented (44%) in this dataset, because Gillick et al. [58] annotated all non-entity mentions (examples in table 5.XI) as /other. We observe that 73% of the wrong

9. It contains roughly 9000 mentions annotated with 88 different types

decisions that our model made on ONTONOTES are committed on this type. In WiFiNE, `/other` always refers to an entity mention, and in most cases the mention has an additional level two and three labels.

5.4 Conclusion

In the first part of this chapter, we perform a qualitative analysis of WiNER as a training data by comparing our corpus with those of other works. In addition, we analyze the behavior of various approaches when training data is enhanced by subsets from our corpus. More precisely, we evaluated the impact of WiNER our corpus on 4 reference NER systems with 6 different NER benchmarks. The LSTM-CRF system of [78] seems to be the one that benefits the most from WiNER overall. Still, shortage of memory or lengthy training times prevent us from measuring the full potential of our corpus. Thus, we proposed an entity-type classifier that exploits a set of features computed over an arbitrary large part of WiNER. Using this classifier for labelling the types of segments identified by a reference NER system yields a 2-stage process that further improves overall performance. WiNER and the classifier we trained are available at <http://rali.iro.umontreal.ca/rali/en/winer-wikipedia-for-ner>.

In the second part of this chapter, we evaluated the impact of WiFiNE on a neural network tagging system based on 2 human made benchmarks. Experiments shows state-of-the-art performances on both benchmarks, when WiFiNE is used as training material. Our analysis on both datasets leads to the following observations. First, enriching Wikipedia articles with proper names, nominal and pronominal mentions systematically leads to better performance, which validates our 4-step approach. Second, the correlation between the train and test type distribution is an important factor to entity typing performance. Third, models could benefit from an example selection strategy based on the genre of the test set.

CHAPTER 6

ROBUST LEXICAL FEATURES FOR IMPROVED NEURAL NETWORK NAMED-ENTITY RECOGNITION

6.1 Overview

In the previous chapter, we explored the typical use of distant supervision data as training material for supervised models. However, the emerging of representation learning techniques have open door for new usages of any sort of data. One can use the data to learn word representations on massive amount of data, then to plug these features in a supervised model for any NLP task. While most research focus on unsupervised word representation learning, we were curious about the ability to learn useful representations from distant supervision data. This chapter is an endeavor to learn a special type of classic word embeddings (such as word2vec [104]) from entity type annotation of WiFiNE. The content of this chapter was published in:

“Abbas Ghaddar and Phillippe Langlais. Robust Lexical Features for Improved Neural Network Named-Entity Recognition. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1896–1907, 2018”

Neural network approaches to Named-Entity Recognition (NER) reduces the need for carefully hand-crafted features. While some features do remain in state-of-the-art systems, lexical features have been mostly discarded, with the exception of gazetteers. In this work, we show that that this is unfair: lexical features are actually quite useful. We propose to embed words and entity types into a low-dimensional vector space we train from annotated data produced by distant supervision. From this, we compute offline a feature vector for representing each word. When used with a vanilla RNN model, this representation yields substantial improvements. We establish a new state-of-the-art F1 score of 87.63 on ONTONOTES 5.0, while remaining very competitive on the CONLL-2003 dataset.

6.2 Introduction

Named-Entity Recognition (NER) is the task of identifying textual mentions and classifying them into a predefined set of types. Various approaches have been proposed to tackle the task, from hand-crafted feature-based machine learning models like conditional random fields [50] and perceptron [132], to deep neural models [33, 98, 151].

Word representations [104, 157], also known as word embeddings, are a key element for multiple NLP tasks including NER [33]. Due to the small amount of reference named-entity annotated data, embeddings are used to extend, rather than replace, hand-crafted features in order to obtain state-of-the-art performance [78].

Recent studies [147, 168] have explored methods for supplying deep sequential taggers with complementary features to standard embeddings. Peters et al. [119] and Tran et al. [156] tested special embeddings extracted from a neural language model (LM) trained on a large corpus. LM embeddings capture context-dependent aspects of word meaning using future (forward LM) and previous (backward LM) context words. When this information is added to standard features, it leads to significant improvements in NER. Also, Chiu and Nichols [30] showed that external knowledge resources (namely gazetteers) are crucial to NER performance. Gazetteer features encode the presence of word n -grams in predefined lists of NEs.

In this work, we discuss some of the limitations of gazetteer features and propose an alternative lexical representation that is trained offline and that can be added to any neural NER system. In a nutshell, we embed words and entity types into a joint vector space by leveraging a large amount of automatically annotated mentions with entity types (120 labels). From this vector space, we compute for each word a 120-dimensional vector, where each dimension encodes the similarity of the word with an entity type. We call this vector an LS representation, for Lexical Similarity. When included in a vanilla LSTM-CRF NER model, LS representations lead to significant gains. We establish a new state-of-the-art F1 score of 87.63 on ONTONOTES 5.0, and obtain near state-of-the-art performance on the CONLL-2003 dataset, without making use of LM embeddings as features.

We first motivate our work in Section 6.3. We present how we compute our LS vectors in Section 6.4. We describe our system in Section 6.5 and report results in Section 6.6. In Section 6.7 we discuss related works, before concluding in Section 7.7.

6.3 Motivation

Gazetteers are lists of entities that are associated with specific NE categories. They are widely used as a feature source in NER, and have been successfully included in feature-based [132] and neural [30] models. Typically, lists of entities are compiled from structured data sources such as DBpedia [9] or Freebase [20]. The surface form of the title of a Wikipedia article, as well as aliases and redirects are mapped to an entity type using the `object_type` attribute of the related DBpedia (or Freebase) page.

Text	Hayao	Tada	,	commander	of	the	Japanese	North	China	Area	Army
LOC	-	-	-	-	-	B	E	-	S	-	-
MISC	-	-	-	S	B	B	E	S	S	S	S
ORG	-	-	-	-	-	B	E	B	I	I	E
PERS	B	E	-	-	-	-	-	-	S	-	-

Image: Chiu and Nichols, 2016

Binary Representation

Figure 6.1 – An example from Chiu and Nichols [30] that show the limitation of binary encoded gazetteer features. For instance, the word *China* appear as entry in 4 lists of named entities, in real world it mostly refer to the country. However, binary features evenly attribute the same weight for the 4 classes. source [30]

Ratinov and Roth [132] use this methodology to compile 30 lists of fine-grained entity types extracted from Wikipedia, while Chiu and Nichols [30] create 4 gazetteers that map to CoNLL categories (PER, LOC, ORG and MISC). Despite their importance, gazetteer-based features suffer from a number of limitations.

- **Binary representation.** As shown in Figure 6.1, gazetteer features encode only the presence of an n -gram in each list and omit its relative frequency. For example, the word « France » can be used as a person, an organization, or a location,

while it likely refers to the country most of the time. Binary features cannot capture this preference.

- **Generation.** At test time, we need to match every n -gram (up to the length of the longest lexicon entry) in a sentence against entries in the lexicons, which is time consuming. In their work, Chiu and Nichols [30] use 4 lists that count over 2.3M entries.
- **Non-entity words.** Gazetteer features do not capture signal from non-entity words, while earlier feature-based models strived to encode that some words (or n -grams) trigger specific entity types. For instance, words such as « eat », « directed » or « born » are words that typically appear after a mention of type PER.

To overcome these limitations, we propose an alternative approach where we embed annotations mined from Wikipedia into a vector space from which we compute a feature vector that represents a word. This vector compactly and efficiently encodes both gazetteer and lexical information. Note that at test time, we only have to feed our model this feature vector, which is efficient.

6.4 Our Method

6.4.1 Corpus Description

In this work, we use WiFiNE, described in Chapter 4 as the source of annotations. In addition, we augmented the corpus with automatically annotated with 7 temporal and numeric entity types (such as Currency, Ordinal, etc...). The overall consists of 157.4M entity mentions that are labelled with 120 fine-grained entity type following using FIGER [91] annotation scheme.

6.4.2 Embedding Words and Entity Types

We used this very large quantity of automatically annotated data for jointly embedding words and entity types into the same low-dimensional space. The key idea consists in learning an embedding for each entity type using its surrounding words. For instance,

the embedding for `/product/software` will be trained using context words that surround all entities that were (automatically) labelled as `/product/software` in Wikipedia. In practice, we found that simply concatenating a sentence (v1) with its annotated version (v2), as illustrated in Figure 6.2, offers a simple but efficient way of combining words and entity types so that embeddings can make good use of them.

<p>(v1) On October 9, 2009, the Norwegian Nobel Committee announced that Obama had won the 2009 Nobel Peace Prize.</p> <p>(v2) On <code>/date</code>, the <code>/organization/government_agency</code> announced that <code>/person/politician</code> had won the <code>/award</code>.</p>
--

Figure 6.2 – Example of the two variants of a given sentence.

We use the FastText toolkit [19] to learn the uncased embeddings for both words and entity types. We train a skipgram model to learn 100-dimensional vectors with a minimum word frequency cutoff of 5, and a window size of 5. This configuration (recommended by the authors) performs the best in the experiments described in Section 6.6. Since FastText learns representations of character n -grams, it has the ability to produce vectors for unknown words.

Figure 6.3 illustrates a T-SNE [158] two-dimensional projection of the embedding of 6 entity types and a sample of 1500 words. Entity type embeddings are marked by big Xs, while circles indicate words. For visualization purposes, we only plot single-word mentions that were annotated with one of those 6 types. Words were randomly and proportionally sampled according to the frequency of each entity type. In addition, words have the color associated with the most frequent type they were annotated with in our resource.

We observe that mentions often annotated by a given type in our resource tend to cluster around this entity type. For instance, « `firefox` » is close to the type `/product/software`, while « `enzyme` » is close to the `biology` entity type. We also notice that words that are labelled with different types tend to appear between types they were annotated with. For instance, « `gpx2` », which is used both as a software and as a gene, has its embedding appear in between `/product/software` and `/biology`.

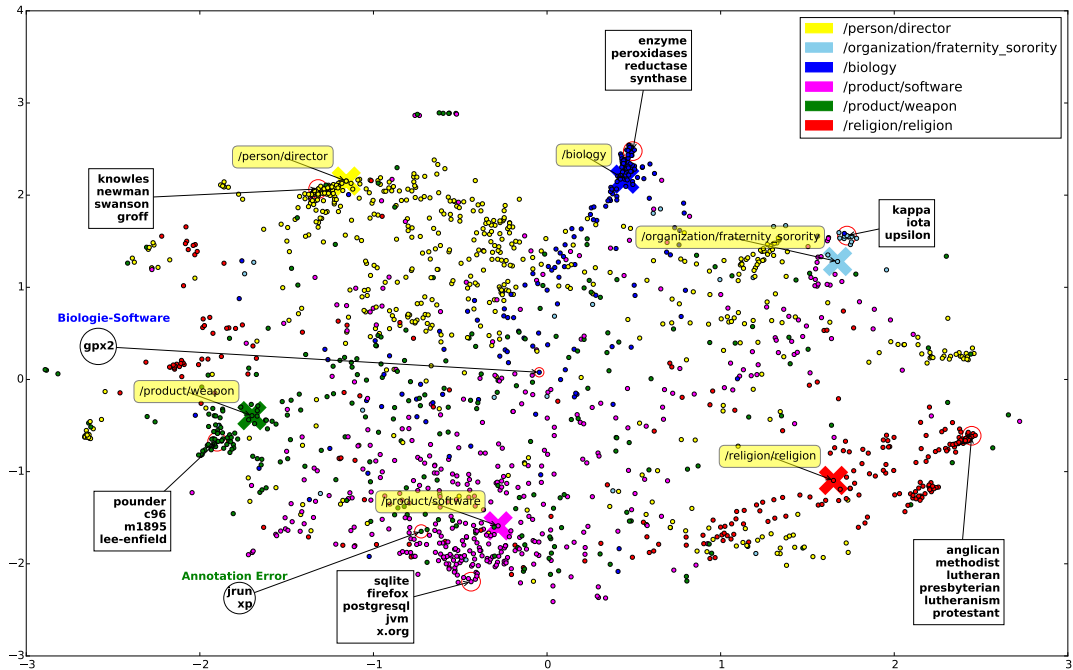


Figure 6.3 – Two-dimensional representation of the vector space which embeds both words and entity types. Big Xs indicate entity types, while circles refer to words (i.e. named entities, here).

We inspected some of the words plotted in Figure 6.3, and found that « jrwn » and « xp » were incorrectly labelled as /product/weapon during the annotation process. But since these words are seen in a *software* context, their embeddings are closer to the /product/software embedding than the /product/weapon one. We feel this tolerance to noise is a desirable feature, one that allows a better distant supervision.

Last, we also observe the tendency of rare words to cluster around their entity type. For instance, « iota » and « x.org » are embedded near their respective types, despite the fact that they appear less than 30 times in the version of Wikipedia we use.

6.4.3 LS Representation

This joint vector space only serves the purpose of associating to each word a LS representation. A LS representation is a 120-dimensional vector where the i th coefficient

is a value in the $[-1, +1]$ interval, equal to the cosine similarity¹ between the word embedding and the embedding of the i th entity type (we have 120 types).

Word	Entity Type	Similarity
hilton	/building/hotel	0.58
	/building/restaurant	0.46
	/person/actor	0.37
gpx2	/biology	0.69
	/product/software	0.56
jrun	/product/software	0.64
	/product/weapon	0.23
dammstadt	/location/city	0.45
	/location/railway	0.44
located	/location	0.47
	/location/city	0.44
directed	/person/director	0.60
	/art/film	0.55
in	/date	0.58
	/location/city	0.54
won	/award	0.53
	/event/sports_event	0.53

Table 6.I – Topmost similar entity types to a few single-word mentions (first four) and non-entity words (last four).

Table 6.I shows the topmost similar entity types for proper names (first four rows) and common words (last four rows). We observe that ambiguous mentions (those annotated with several types) are adequately handled. For instance, the LS representation of the word « hilton » encodes that it more often refers to a hotel or a restaurant than to an actress. Also, we observe that entity words that are either not or rarely annotated in our resource are still adequately associated with their right type. For instance, « dammstadt », which appears only 5 times in our corpus, and which refers to the Damm city in Germany, is most similar to `/location/city` and `/location/railway`. Interestingly, it turns out that this mention does not have its page in English Wikipedia.

Furthermore, we observe that non-entity context words have a strong similarity to

1. The cosine similarity outperforms other metrics in our experiments.

types they precede or succeed. For instance the verb « directed » is very close to `/person/director`, an entity type that usually precedes it, and to `/art/film`, that usually follows it. Likewise, the preposition « in » is near `/date` and `/location/city`, which frequently follow "in".

6.4.4 Strength of the LS Representation

To summarize, we propose a compact lexical representation which is computed offline, therefore incurring no computation burden at test time. This representation is tolerant to the inherent noise of distant supervision. It encodes the preference of an entity-mention word for a given type, an information out of reach from binary gazetteer features. It also lends itself nicely to the inclusion of lexical features that have been successfully used in earlier feature-based systems [97, 132]. Also, because entity types are well represented in our resource, their embeddings are robust: Our representation does accommodate unfrequent words.

6.5 Our NER System

In order to test the efficiency of our lexical feature representation, we implemented a state-of-the-art NER system we now describe.

6.5.1 Bi-LSTM-CRF Model

We adopt the popular Bi-LSTM-CRF architecture (Figure 6.4), a *de facto* baseline in many sequential tagging tasks [78, 147, 151].

6.5.2 Features

In addition to the LS vector, we incorporate publicly available pre-trained embeddings, as well as character-level, and capitalization features. Those features have been shown to be crucial for state-of-the-art performance.

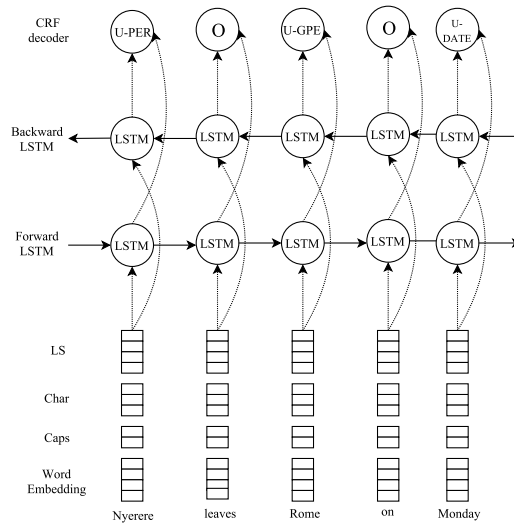


Figure 6.4 – Main architecture of our NER system.

6.5.2.1 Word Embeddings

We experimented with several publicly available word embeddings, such as Senna [33] Word2Vec [104], GloVe [118], and SSKIP [175]. We find that the latter performs the best in our experiments. SSKIP embeddings are 100-dimensional case sensitive vectors that were trained using a n -skip-gram model [175] on 42B tokens. These embeddings were previously used by [78, 151], who report good performance on CONLL, and state-of-the-art results on ONTONOTES 5.0 respectively. Note that these pre-trained embeddings are adjusted during training.

6.5.2.2 Character Embeddings

Following [78], we use a forward and a backward LSTM to derive a representation of each word from its characters. A character lookup table is randomly initialized, then trained at the same time as the Bi-LSTM model sketched in Section 6.5.1. Figure 6.5 illustrates our architecture to generate the character-level representation of each word. First, a character lookup table is randomly initialized. We use a forward and a backward LSTM to derive a representation of each word from its characters. This approach has

been used in the recent work of [78].

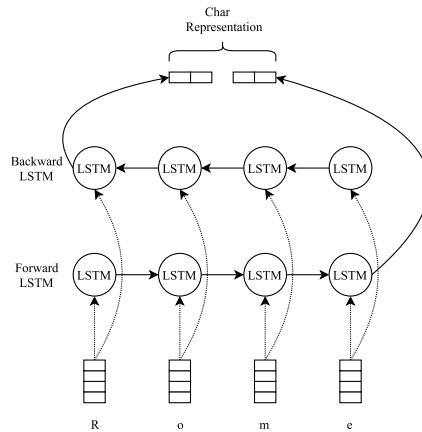


Figure 6.5 – Character representation of the word «Roma» given to the bidirectional LSTM of Figure 6.4.

6.5.2.3 Capitalization Features

Similarly to previous works, we use capitalization features for characterizing certain categories of capitalization patterns: `allUpper`, `allLower`, `upperFirst`, `upperNotFirst`, `numeric` or `noAlphaNum`. We define a random lookup table for these features, and learn its parameters during training.

6.5.2.4 LS Features

Contrarily to previous features, lexical vectors are computed offline and are not adjusted during training. We found useful in practice to apply a `MinMax` scaler in the range $[-1, +1]$ to each LS vector we computed; thus, $[.., 0.095, .., 0.20, .., 0.76, ..]$ becomes $[.., -1, .., -0.67, .., 1, ..]$.

6.6 Experiments

6.6.1 Data and Evaluation

We consider two well-established NER benchmarks: CONLL-2003 and ONTONOTES 5.0. Table 6.II provides an overview of the two datasets. As we can see, ONTONOTES is much larger. For both datasets, we convert the IOB encoding to BILOU, since previous work [132] found the latter to perform better. In keeping with others, we report mention-level F1 score using the `conlleva1` script².

The CONLL-2003 NER dataset [154] is a well known collection of Reuters newswire articles that contain a large portion of sports news. It is annotated with four entity types: *Person* (PER), *Location* (LOC), *Organization* (ORG) and *Miscellaneous* (MISC). The four entity types are fairly evenly distributed, and the train/dev/test datasets present a similar type distribution.

The ONTONOTES 5.0 dataset [66, 124] includes texts from five different genres: broadcast conversation (200k), broadcast news (200k), magazine (120k), newswire (625k), and web data (300k). This dataset is annotated with 18 entity types, and it's much larger than CONLL. Following previous researches [30, 151], we use the official train/dev/test split of the CoNLL-2012 shared task [123]. Also, we exclude (both during training and testing) the New Testaments portion as it does not contains gold NE annotations.

6.6.2 Training and Implementation

Training is carried out by mini-batch stochastic gradient descent (SGD) with a momentum of 0.9 and a gradient clipping of 5.0. The mini-batch is 10 for both datasets, and learning rates are 0.009 and 0.013 for CONLL and ONTONOTES respectively. More sophisticated optimization algorithms such as AdaDelta [176] or Adam [75] converge faster, but none outperformed SGD in our experiments.

Our system uses a single Bi-LSTM layer at the word level whose hidden dimensions are set to 128 and 256 for CONLL and ONTONOTES respectively. For both models, the character embedding layer is 25, and the hidden dimension of the forward and backward

2. <http://www.cnts.ua.ac.be/conll2000/chunking/conlleva1.txt>

Dataset		Train	Dev	Test
CONLL	<i>#tok</i>	204,567	51,578	46,666
	<i>#ent</i>	23,499	5,942	5,648
ONTO	<i>#tok</i>	1,088,503	147,724	152,728
	<i>#ent</i>	81,828	11,066	11,257

Table 6.II – Statistics of the CONLL-2003 and ONTONOTES 5.0 datasets. *#tok* stands for the number of tokens, and *#ent* indicates the number of named-entities gold annotated.

character LSTMs are set to 50. To mitigate overfitting, we apply a dropout mask [148] with a probability of 0.5 on the input and output vectors of the Bi-LSTM layer. For both datasets, we set capitalization embeddings to 25 and trained the models up to 50 epochs.

We tuned the hyper-parameters by grid search, and used early stopping based on the performance on the development set. We varied dropout ([0.25, 0.5, 0.65]), hidden units ([50, 128, 256, 300]), capitalization ([10, 20, 30]) and chars ([25, 50, 100]) embedding dimensions, learning rate ([0.001, 0.015] by step 0.002), and optimization algorithms and fixed the other hyper-parameters. We implemented our system using the Tensorflow [1] library, and ran our models on a GeForce GTX TITAN Xp GPU. Training requires about 1.5 hours for CONLL and 5 hours for ONTONOTES.

6.6.3 Results on Dev

Table 6.III shows the development set performance of our final models on each dataset compared with the work of Chiu and Nichols [30]. The authors use a architecture similar to ours, but use a binary gazetteer feature set, while we use our LS representation. Since our systems involve random initialization, we report the mean as well as the standard deviation over five runs. The improvements yielded by our model on the CONLL dataset are significant although modest, while those observed on ONTONOTES are more substantial. We also observe a lower variance of our system over the 5 runs.

	CONLL	ONTO
C&N. 2016	94.03 (± 0.23)	84.57 (± 0.27)
Our model	94.53 (± 0.10)	86.20 (± 0.14)

Table 6.III – Development set F1 scores of our best hyper-parameter setting compared to the results reported by [30].

6.6.4 Results on CONLL

Table 6.IV reports the our model’s performance³ on the CONLL-2003 test set, as well as the performance of systems previously tested on this test set (the figures are those published by the authors). Because of the small size of the training set, some authors [30, 119, 168] incorporated the development set as a part of training data after tuning the hyper-parameters. Consequently, their results are not directly comparable, so we do not report them.

First, we observe that our model significantly outperforms models that use extensive sets of hand-crafted features [88, 132] as well as the the system of [97] that uses NE and Entity Linking annotations to jointly optimize the performance on both tasks. Second, our model outperforms as well other NN models that only use standard word embeddings, which indicates that our lexical feature vector is complementary to standard word embeddings.

Third, our system delivers competitive performance with state-of-the-art models that use a more complex architecture and more elaborate features. Tran et al. [156] use three layers of Stacked Residual RNN (Bi-LSTM) with bias decoding. Our model is much simpler and faster. They report a performance of 90.43 when using an architecture similar to ours. The two systems that have slightly higher F1 scores on the CONLL dataset both use embeddings obtained from a forward and a backward Language Model trained on the One Billion Word Benchmark [25]. They report gains between 0.8 and 1.2 points by using such LM embeddings, which suggests that LS lexical vectors are indeed efficient. Unfortunately, due to time and resource constraints⁴, we were not able

3. Standard deviation on the test set is reported in Table 6.VII due to space constraints in Table 6.IV

4. LM embeddings are not publicly available, and according to Jozefowicz et al. [71], they required

Model	L G C E c M S	F1
Finkel et al. [50]	+ + + • • • •	86.86
Ratinov and Roth [132]	+ + + • • • •	90.88
Lin and Wu [88]	+ + + • • • •	90.90
Luo et al. [97]	+ + + • • • •	91.20
Collobert et al. [33]	• + + + • • •	89.56
Huang et al. [69]	• • + + + • •	90.10
Lample et al. [78]	• • + + + • •	90.94
Ma and Hovy [98]	• • + + + • •	91.21
Shen et al. [142]	• • + + • • •	90.89
Strubell et al. [151]	• • + + • • •	90.54
Tran et al. [156]	• • + + + + •	91.69
Liu et al. [93]	• • + + + + •	91.71
This work	• + + + + • +	91.61

Table 6.IV – F1 scores on the CONLL test set. The first four systems are feature-based, the others are neuronal. The LGCEcMS column indicates the feature configuration of each system. L stands for Lexical feature, G for Gazetteers, C for Capitalization, E for pre-trained Embeddings, c for character embeddings, M for language Model Embeddings, and S for the proposed LS feature representation. + indicates that the model use the feature set.

Model	BC	BN	MZ	NW	TC	WB
Finkel and Manning [49]	78.66	87.29	82.45	85.50	67.27	72.56
Durrett and Klein [45]	78.88	87.39	82.46	87.60	72.68	76.17
Chiu and Nichols [30]	85.23	89.93	84.45	88.39	72.39	78.38
This work	86.25	90.41	85.87	89.67	75.41	80.39

Table 6.V – Per-genre F1 scores on ONTONOTES (numbers taken from Chiu and Nichols [30]). BC = broadcast conversation, BN = broadcast news, MZ = magazine, NW = newswire, TC = telephone conversation, WB = blogs and newsgroups.

to measure whether both features complement each other.

6.6.5 Results on ONTONOTES

Table 6.VI reports the F1 score of our system compared to the performance reported by others on the ONTONOTES test set. To the best of our knowledge, we surpass previ-

three weeks to train on 32 GPUs.

ously reported F1 scores on this dataset. In particular, our system significantly outperforms the Bi-LSTM-CNN-CRF models of [30] and [151] by an absolute gain of 1.35 and 0.64 points respectively. Less surprisingly, it surpasses systems with hand-crafted features, including Ratnov and Roth [132] that use gazetteers, and the system of Durrett and Klein [45] which uses coreference annotation in ONTONOTES to jointly model NER, entity linking, and coreference resolution tasks.

Model	L G C E c M S	F1
Finkel and Manning [49]	+ + + ● ● ● ●	82.42
Ratnov and Roth [132]	+ + + ● ● ● ●	84.88
Passos et al. [115]	+ + + ● ● ● ●	82.24
Durrett and Klein [45]	+ + + ● ● ● ●	84.04
Chiu and Nichols [30]	● + + + + ● ●	86.28
Shen et al. [142]	● ● + + + ● ●	86.52
Strubell et al. [151]	● ● + + + ● ●	86.99
This work	● + + + + ● +	87.63

Table 6.VI – F1 scores on the ONTONOTES test set. The first four systems are feature-based, the following ones are neuronal. See Table 6.IV for an explanation of the LGCEcMS column.

The ONTONOTES benchmark is annotated with 18 types (e.g. LAW, PRODUCT) and contains many rare words, especially in the Web data collection. Chiu and Nichols [30] note that the 4-class gazetteer they used yielded marginal improvements on ONTONOTES, contrarily to CONLL. In particular, they observe that mentions that match LOC entries in their gazetteer often match GPE, NORP and lists. They suggest that a finer-grained gazetteer could improve the performance of their system on ONTONOTES. Our results confirm this, since we use 120 types.

We further detail the gains we observed for each sub-collection of texts in the test set. Table 6.V reveals that major improvements over [30]’s model are on noisier collections such as telephone conversations (+3 points) and blogs or newsgroups (+2 points). Those type of texts are characterized by a large set of infrequent words, for which classical embeddings are typically poorly trained. Our approach does not seem to suffer from this problem as severely, as discussed in Section 6.3.

6.6.6 Ablation Results

In this experiment, we directly compare the LS representation with the SSKIP word-embedding feature set. In order to maintain a high level of performance, both character and capitalization features are used in all configurations. We want to point out that LS vectors are not adapted during training, contrarily to the SSKIP embeddings. Similarly to Section 6.6.3, we report in Table 6.VII, for each feature configuration, the average F1 score as well as the standard deviation over five runs.

We observe that on both CONLL and ONTONOTES, the SSKIP model outperforms our feature vector approach by 0.65 F1 points on average. The difference is not as high as we first expected, especially since the SSKIP model is adjusted during training, while our representation is not. Still, LS representations seem to encode a large portion of the information needed to model the NER task. Also, it is worth mentioning that our embeddings are trained on 1.3B words compared to 42B for SSKIP.

Model	CONLL	ONTONOTES
SSKIP	90.52 (\pm 0.18)	86.57 (\pm 0.10)
LS	89.94 (\pm 0.16)	85.92 (\pm 0.12)
all	91.61 (\pm 0.10)	87.63 (\pm 0.13)

Table 6.VII – F1 scores of differently trained systems on CONLL-2003 and ONTONOTES 5.0 datasets. Capitalization (Section 6.5.2.3) and character features (Section 6.5.2.2) are used by default by all models.

We also observe that models that use both feature sets significantly outperform other configurations. To confirm that the gains came from our feature vector and not from increasing the number of hidden units, we tested several SSKIP models by increasing the LSTM hidden layer dimension so that number of parameters is the same as the model with LS vectors. We observed a degradation of performance on both datasets, mostly due to overfitting on the training data. From those results, we conclude that our lexical representation and the SSKIP one are complementary.

6.7 Related Works

Traditional approaches to NER, like CRF-based [50] and Perceptron-based systems [132] have dominated the field for over a decade. They rely heavily on hand-engineered features [97] and external resources such as gazetteers. One major drawback of such an approach is its weak generalization power [78].

Current state-of-the-art systems [30, 151] use a combination of Convolutional Neural Networks (CNNs), Bi-LSTMs, along with a CRF decoder. CNNs are used to encode character-level features (prefix and suffix), while LSTM is used to encode word-level features. Finally, a CRF is placed on top of those models in order to decode the best tag sequence. Pre-trained embeddings obtained by unsupervised learning are core features of these models. In this work, we show that deep NN architectures can also benefit from lexical features, at least when encoded in the compact form we propose.

Tran et al. [156] and Peters et al. [119] propose an alternative semi-supervised approach different from ours. They incorporate LM embeddings that were pre-trained on a large unlabelled corpus as features for NER. These embeddings allow to generate a representation for a word depending on its context. For instance, the LM embeddings of the word *France* in « *France is a developed country* » is different than that in « *Anatole France began his literary career* ». Such embeddings are trained on very large amount of texts. Our feature set is crafted from distant supervision applied to Wikipedia, a much less time-consuming process which we showed to be nevertheless adapted to rare words.

Chiu and Nichols [30] used gazetteer features in order to establish state-of-the-art performance on both CONLL-2003 and ONTONOTES5.0. They mined DBPedia in order to compile 4 lists of named-entities that contain over 2.3M entries. We show that LS representations outperform their gazetteer features.

6.8 Conclusion

In this chapter, we have explored the idea of generating lexical features (considered seen as classic word embeddings) for NER out of distant supervision data. We used WiFiNE to train a vector space that jointly embeds words and named-entities. This vec-

tor space is used to compute a 120 dimensional vector per word, which encodes the similarity of the word to each of the entity types. Our results suggest that our proposed lexical representation, even though it is not adjusted at training time, provides very competitive results compared to more complex approaches on the well-studied CONLL dataset, and delivers a new state-of-the-art F1 score of 87.54 on the more diversified ONTONOTES dataset. We further observe larger gains on collections with more unfrequent words. The source code and the data we used in this work are publicly available at <http://rali.iro.umontreal.ca/rali/en/wikipedia-lex-sim>, with the hope that other researchers will report gains, when using our lexical representation. As a future work, we want to investigate the usefulness of our LS feature representation on other NER tasks, including NER in tweets where out-of-vocabulary and low-frequency words represent a challenge.

CHAPTER 7

CONTEXTUALIZED WORD REPRESENTATIONS FROM DISTANT SUPERVISION WITH AND FOR NER

7.1 Overview

In this Chapter, we describe a special type of deep contextualized word representation that is *learned from* distant supervision annotations and *dedicated to* named entity recognition. Our extensive experiments on 7 datasets show systematic gains across all domains over strong baselines, and demonstrate that our representation is complementary to previously proposed embeddings. Furthermore, we show that simply stacking various representations significantly boost performances. By doing so, we obtained new state-of-the-art results on two well-established datasets: CONLL-2003 and ONTONOTES 5.0: 0.4 and 1.1 absolute improvements respectively.

7.2 Introduction

Contextualized word representations are nowadays a resource of choice for most NLP tasks [120]. These representations are trained with unsupervised language modelling [71], masked-word prediction [41], or supervised objectives like machine translation [99]. Despite their strength, best performances on downstream tasks [4, 61, 84] are always obtained when these representations are stacked with traditional (classic) word embeddings [104, 118]. This indicates that traditional and contextualized embeddings do not encode the same information; indeed they complement each other. This in turn leaves the door open for further improvements using embeddings obtained with other source of data and objectives.

Our main contribution in this work is to revisit the idea proposed in the previous chapter. Motivated by the recent success of pre-trained language model embeddings, we propose a contextualized word representation trained WiFiNE. We do so by training a model to predict the entity type of each word in a given sequence (e.g. paragraph).

We run extensive experiments feeding our representation, along side with previously proposed traditional and contextualized ones, as features to a vanilla Bi-LSTM-CRF [98]. Results show that our contextualized representation leads to significant boost in performance on 7 NER datasets of various sizes and domains. The proposed representation surpasses the one of Ghaddar and Langlais [57] and is complementary to popular contextualized embeddings like ELMo [120].

By simply stacking various representations, we report new state-of-the-art¹ performances on CoNLL-2003 [154] and ONTONOTES 5.0 [124] with a F1 score of 93.22 and 89.95 respectively.

7.3 Data and Preprocessing

We leverage the entity type annotations in WiFiNE, we use the fine-grained type annotation available in the resource (e.g. `/person/politician`). Also, inspired by the recent success of masked-word prediction [41], we further apply preprocessing to the original annotations by **(a)** replacing an entity by a special token [MASK] with a probability of 0.2, and **(b)** replacing primary entity mentions, e.g. all mentions of *Barack Obama* within its dedicated article, by the special mask token with a probability of 0.5.

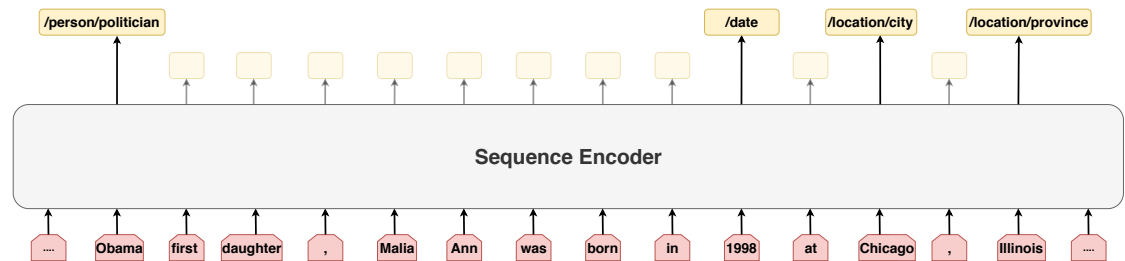


Figure 7.1 – Input (rose) and output (yellow) sequences used by our encoder to learn contextualized representations. Transparent yellow box indicates that no prediction is made for the corresponding token.

In WiFiNE, named-entities that do not have a Wikipedia article (e.g. *Malia Ann* in Figure 7.1) are left unannotated, which introduces false negatives. Therefore, we mask

1. as of January 2019

non-entity words when we calculate the loss. Although contextualized representation learning has access to arbitrary large contexts (e.g. the document), in practice representations mainly depend on sentence level context [24]. To overcome this limitation to some extent, we use the Wikipedia layout provided in WiFiNE to concatenate sentences of the same paragraphs, sections and document up to a maximum size of 512 tokens.

An illustration of the preprocessing is depicted in Figure 7.1 where for the sake of space, a single sentence is shown. Masked entities encourage the model to learn good representations for non-entity words even if they do not participate in the final loss. Because our examples are sections and paragraphs, the model will be forced to encode sentence- as well as document-based context. In addition, training on (longer) paragraphs is much faster and memory efficient than batching sentences.

7.4 Learning our Representation

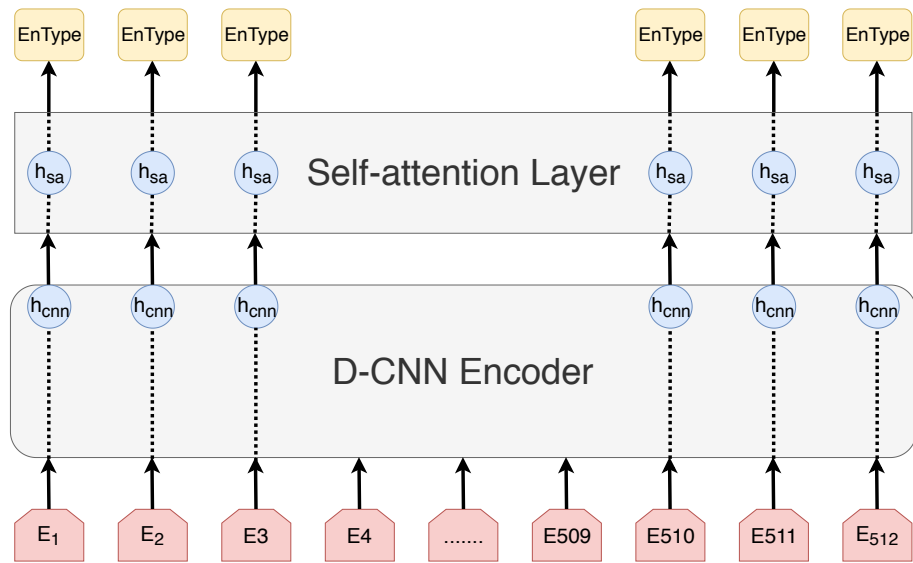


Figure 7.2 – Illustration of the architecture of the model used for learning our representation. It consists of stacked layers of dilated convolutional neural network followed by a self-attention layer. The input is a sequence of tokens with a maximum length of 512, where the output is the associated entity type sequence. We use the hidden state of the last DCNN layer and the self-attention layer as our representation.

We use a model (Figure 7.2) composed of a multi-layer bidirectional encoder that produces hidden states for each token in the input sequence. At the output layer, the last hidden states are fed into a softmax layer for predicting entity types. Following [151], we used as our encoder the Dilated Convolutional Neural Network (DCNN) with an exponential increasing dilated width. DCNN was first proposed by [174] for image segmentation, and was successfully deployed for NER by [151]. The authors show that stacked layers of DCNN that incorporate document context have comparable performance to Bi-LSTM while being 8 times faster. DCNN with a size 3 convolution window needs 8 stacked layers to incorporate the entire input context of a sequence of 512 tokens, compared to 255 layers using a regular CNN. This greatly reduces the number of parameters and makes training more scalable and efficient.

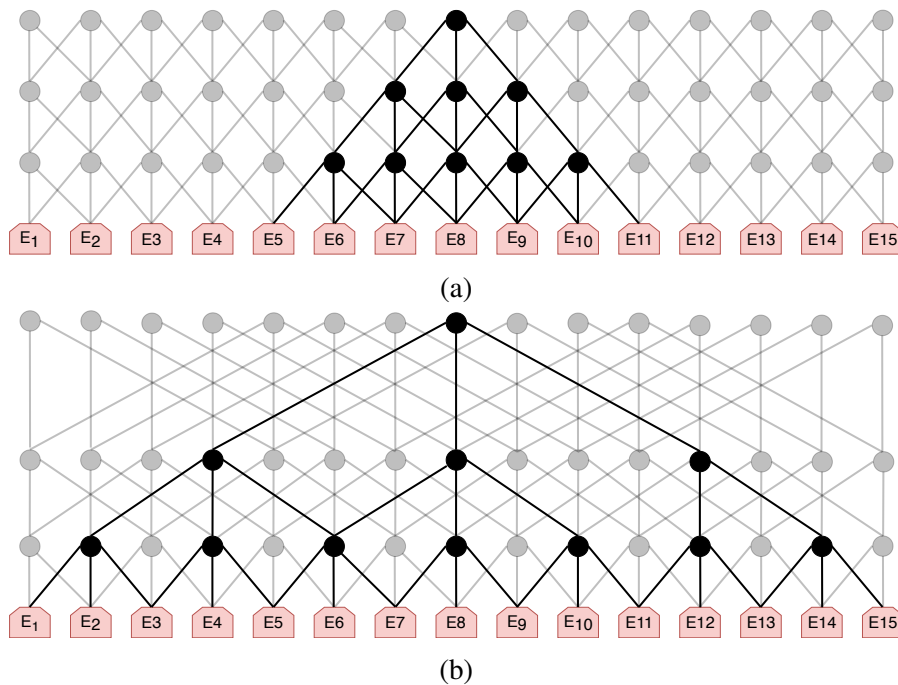


Figure 7.3 – Difference between Regular (sub-figure a) and Dilated (sub-figure b) CNN. Like typical CNN layers, dilated convolutions operate on a sliding window of context over the sequence, but unlike conventional convolutions, the context need not be consecutive.

Because our examples are paragraphs rather than sentences, we employ a self-attention mechanism on top of DCNN output with the aim to encourage the model to focus on

salient global information. In this paper, we adopt the multi-head self-attention formulation by Vaswani et al. [160]. Comparatively, Transformer-based architectures [41] require a much larger² amount of resources and computations. To improve the handling of rare and unknown words, our input sequence consists of WordPiece embeddings [167] as used by Devlin et al. [41], Radford et al. [127]. We use the same vocabulary distributed by the authors, as it was originally learned on Wikipedia.

	John	was	born	in	Montreal
John	0.06	0.12	0.76	0.04	0.02
was	*	*	*	*	*
born	*	*	*	*	*
in	*	*	*	*	*
Montreal	0.08	0.09	0.37	0.41	0.05

Figure 7.4 – An artificial example that illustrate how attention weights of a self-attention layer can behave if the outputs to predict are entity types. For example, the word *born* that precedes *John* is a strong indicator that the latter should be tagged as person. Similarly, for the target word *Montreal* and its context *born in*.

We use 8 stacked layers of DCNN to encode input sequences of maximum length of 512. WordPiece and position embeddings, number of filters in each dilated layer and self-attention hidden units were all set to 384. For self-attention, we use 6 attention heads and set intermediate hidden unit to 512. We apply a dropout mask [148] with a probability of 0.3 at the end of each DCNN layer, and at the input and output of the self-attention layer. We adopt the Adam [75] optimization algorithm, set the initial learning rate to $1e^{-4}$, and use an exponential decay. We train our model up to 1.5 millions steps with mini-batch size of 64. We implemented our system using the Tensorflow [1] library, and training requires about 5 days on a single TITAN XP GPU.

2. Actually prohibitive with our single GPU computer.

7.5 Experiments on NER

7.5.1 Datasets

To compare with state-of-the-art models, we consider two well-established NER benchmarks: CONLL-2003 [154] and ONTONOTES 5.0 [123]. To further determine how useful our learned representation is on other domains, we also considered three additional datasets: WNUT17 [40] (social media), I2B2 [153] (biomedical), and FIN [6] (financial). In addition, we perform an out-domain evaluation for models trained on CONLL-2003 and tested on WIKIGOLD [13] (wikipedia) and WEBPAGES [132] (web pages).

Dataset	Domain	Types	# entities		
			train	dev	test
CONLL	news	4	23499	5942	5648
ONTONOTES	news	18	81828	11066	11257
WNUT17	tweet	6	1975	836	1079
I2B2	bio	23	11791	5453	11360
FIN	finance	4	460	-	120
WIKIGOLD	wikipedia	4	-	-	3558
WEBPAGES	web	4	-	-	783

Table 7.I – Statistics on the datasets used in our experiments.

Table 7.I lists the dataset used in this study domain, label size, and number of mentions in train/dev/test portions. We used the last 2 datasets to perform an out-of-domain evaluation of CONLL models. Those are small datasets extracted from Wikipedia articles and web pages respectively, and manually annotated following CONLL-2003 annotation scheme.

7.5.2 Input Representations

Our NER model is a vanilla Bi-LSTM-CRF [98] that we feed with various representations (hereafter described) at the input layer. Our system is a single Bi-LSTM layer with a CRF decoder, with 128 hidden units for all datasets except for ONTONOTES and

I2B2 where we use 256 hidden units. For each learned representations (ours, ELMo, FLAIR, BERT), we use the weighted sum of all layers as input, where weights are learned during training. For each word, we stack the embeddings by concatenating them to form the input feature of the encoder.

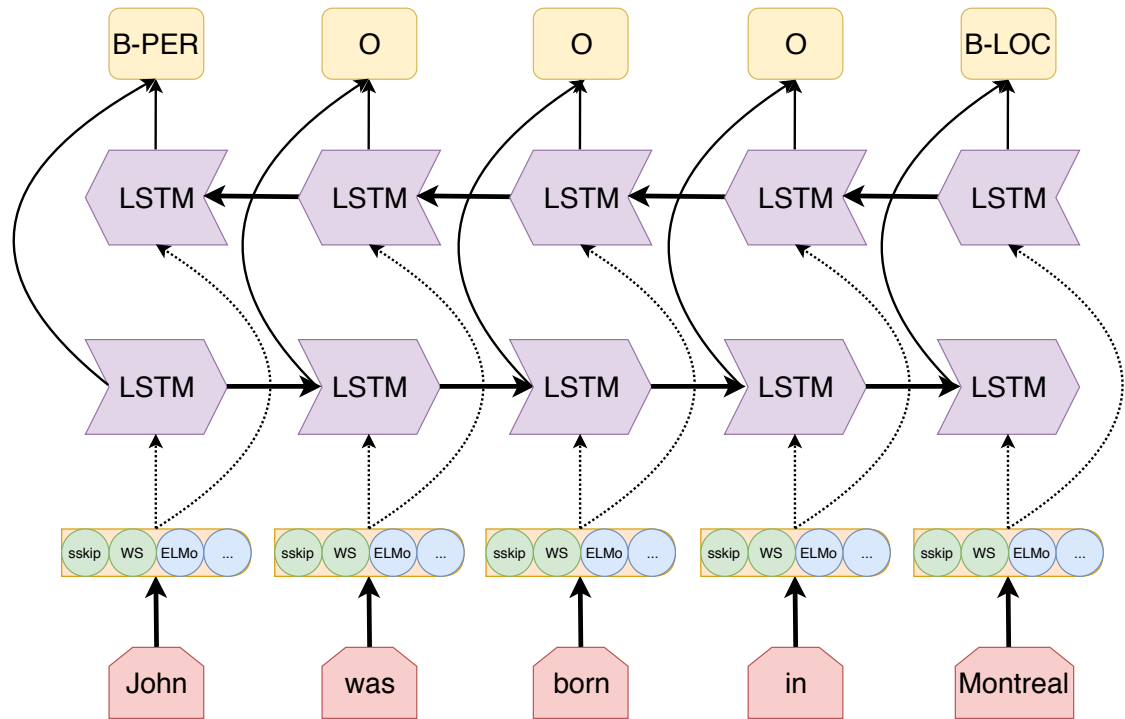


Figure 7.5 – Main architecture of NER model used in this work. Green circle at the input layer show baseline features, while blue ones show contextualized word representations that are added.

Training is carried out by mini-batch of stochastic gradient descent (SGD) with a momentum of 0.9 and a gradient clipping of 5.0. To mitigate over-fitting, we apply a dropout mask with a probability of 0.7 on the input and output vectors of the Bi-LSTM layer. The mini-batch is 10 and learning rate is 0.011 for all datasets. We trained the models up to 63 epochs and use early stopping based on the official development set. For FIN, we randomly sampled 10% of the train set for development.

7.5.2.1 Word-Shape Features

We use 7 word-shape features: `allUpper`, `allLower`, `upperFirst`, `upperNotFirst`, `numeric`, `punctuation` or `noAlphaNum`. We randomly allocate a 25-dimensional vector for each feature, and learn them during training.

7.5.2.2 Traditional Word Embeddings

We use the 100-dimensional case sensitive SSKIP [90] word embeddings. We also compare with the previously described 120-dimensional vector representation of [57], they call it LS.

7.5.2.3 Contextualized Word Embeddings

We tested 3 publicly available contextualized word representations: ELMo [120]: $dim = 1024$, $layers = 3$; FLAIR [4]: $d = 2048$, $l = 1$; and BERT [41]: $d = 1024$, $l = 4$. For the latter, we use the hidden state of the 4 last layers of the `Large` model. For the proposed representation, we use the hidden state of the last DCNN layer and the self-attention layer as feature input ($d = 384$, $l = 2$). Following Peters et al. [120], each representation (including ours) is the weighted sum of the hidden layers, where weights are learned during training. We use concatenation to stack the resulting representations in the input layer of our vanilla Bi-LSTM-CRF model, since [?] show that concatenation performs reasonably well in many NLP tasks.

7.6 Experiments

7.6.1 Comparison to LS embeddings

Since we used the very same distant supervision material for training our contextual representation, we compare it to the one of Ghaddar and Langlais [57]. We concentrate on CONLL-2003 and ONTONOTES 5.0, the datasets most often used for benchmarking NER systems.

	Conll			Ontonotes		
	\mathcal{X}	LS	ours	\mathcal{X}	LS	ours
ws+sskip	90.37	91.23 (+0.9)	91.76 (+1.4)	86.44	87.95 (+0.9)	88.13 (+0.9)
ws+sskip+elmo	92.47	92.49 (+0.0)	92.82 (+0.4)	89.37	89.44 (+0.1)	89.68 (+0.3)
ws+sskip+elmo+flair	92.69	92.75 (+0.1)	93.22 (+0.5)	89.55	89.59 (+0.0)	89.73 (+0.2)
ws+sskip+elmo+flair+bert	92.91	92.87 (+0.0)	93.01 (+0.1)	89.66	89.70 (+0.0)	89.95 (+0.3)
Peters et al. [120]		92.20			-	
Clark et al. [31]		92.61			88.81	
Devlin et al. [41]		92.80			-	

Table 7.II – F1 scores over five runs on CONLL and ONTONOTES test set of ablation experiments. We evaluate 4 baselines without additional embeddings (column \mathcal{X}) and with LS embeddings [57] or ours. Figures in parenthesis indicate the gain over the baselines.

Table 7.II reports results of 4 strong baselines that use popular embeddings (column \mathcal{X}), further adding either the LS representation [57] or ours. In all experiments, we report the results on the test portion of models performing best on the official development set of each dataset. As a point of comparison, we also report 2018 state-of-the-art systems.

First we observe that adding our representation to all baseline models leads to systematic improvements, even for the very strong baseline which exploits all three contextual representations (fourth line). The LS representation does not deliver such gains, which demonstrates that our way of exploiting the very same distant supervision material is more efficient. Second, we see that adding our representation to the weakest baseline (line 1), while giving a significant boost, does not deliver as good performance as when adding other contextual embeddings. Nevertheless, combining all embeddings yields state-of-the-art on both CONLL and ONTONOTES.

7.6.2 Comparing Contextualized Embeddings

Table 7.III reports F1 scores on the test portion of the 7 datasets we considered, for models trained with different embedding combinations. Our baseline is composed of word-shape and traditional (SSKIP) embeddings. Then, contextualized word representations are added greedily, that is, the representation that yields the largest gain when

considered is added first and so forth.

Expectedly, ELMo is the best representation to add to the baseline configuration, with significant F1 gains for all test sets. We are pleased to observe that the next best representation to consider is ours, significantly outperforming FLAIR. This is likely due to the fact that both FLAIR and ELMo embeddings are obtained by training a language model, therefore encoding similar information.

	In Domain					Out Domain	
	Conll	Onto	WNUT	FIN	I2B2	WikiGold	WebPage
WS+SSKIP	90.73	86.44	32.30	81.82	86.41	66.03	45.13
+ELMo	92.47	89.37	44.15	82.03	94.47	76.34	54.45
+Ours	92.96	89.68	47.40	83.00	94.75	78.51	57.23
+FLAIR	93.22	89.73	46.80	83.11	94.79	77.77	56.20
+BERT	93.02	89.97	46.47	81.94	94.92	78.06	56.84

Table 7.III – Mention-level F1 scores. The baseline (first line) uses word shape and traditional (classic) embeddings. Variants stacking various representations are presented in decreasing order of F1 return. So for instance, ELMo is the best representation to add to the baseline one.

Continuously aggregating other contextual embeddings (FLAIR and BERT) leads to some improvements on some datasets, and degradations on others. In particular, stacking all representations leads to the best performance on 2 datasets only: ONTONOTES and I2B2. Those datasets are large, domain diversified, and have more tags than other ones. In any case, stacking word-shapes, SSKIP, ELMo and our representation leads to a strong configuration across all datasets. Adding our representation to ELMo, actually brings noticeable gains (over 2 absolute F1 points) in out-domain settings, a very positive outcome.

Surprisingly, BERT did not perform as we expected, since they bring minor (ONTONOTES) or no (CONLL) improvement. We tried to reproduce the results of fine-tuned and feature-based approaches reported by the authors on CONLL, but as many others,³ our results were disappointing.

3. <https://github.com/google-research/bert/issues?utf8=%E2%9C%93&q=NER>

7.6.3 Analysis

	Tok_1	Tok_2	Tok_3	Tok_510	Tok_511	Tok_512	
Tok_1	*	*	*	*	*	*	*	→ 2 times
Tok_2	*	*	0.69	*	*	0.78	0.71	→ 41 times
Tok_3	*	0.59	*	*	0.67	*	*	→ 37 times
.....	*	*	*	*	*	*	*	
Tok_510	*	*	*	*	*	*	*	→ 1 times
Tok_511	*	*	*	*	*	*	*	→ 2 times
Tok_512	*	*	*	*	*	*	*	→ 0 times

Figure 7.6 – An illustration on how we analyze the impact of self-attention in influencing document context. We check the words that received the highest attention weights inside each document for CONLL dev portion.

We suspect one reason for the success of our representation is that it captures document wise context. As shown in Figure 7.6, we inspected the words the most attended according to the the self-attention layer of some documents, an excerpt of which is reported in Figure 7.7. We observe that attended words in the document are often related to the topic of the document.

84	economic	<i>Stock, mark, Wall, Treasury, bond</i>
148	sport	<i>World, team, record, game, win</i>
201	news	<i>truck, Fire, store, hospital, arms</i>

Figure 7.7 – top 5 attended words for some randomly picked documents in the dev set of CONLL. Column 1 indicate document number, while column 2 is our appreciation of the document topic.

We further checked whether the gain could be imputable to the fact that WiFiNE contains the mentions that appear in the test sets we considered. While this of course happens (for instance 38% of the test mentions in ONTONOTES are in the resource), the performance on those mentions with our representation is no better than the performance

on other mentions.

7.7 Conclusion

We have explored the idea of generating a contextualized word representation from distant supervision annotations coming from Wikipedia, improving over the LS static representation we proposed in Chapter 6. When combined with popular contextual ones, our representation leads to state-of-the-art performance on both CONLL and ONTONOTES. We are currently analyzing the complementarity of our representation to others.

We plan to investigate tasks such as coreference resolution and non-extractive machine reading comprehension, where document level context and entity type information is crucial. The source code and the pre-trained models we used in this work are publicly available at <http://rali.iro.umontreal.ca/rali/en/wikipedia-ds-cont-emb>

CHAPTER 8

CONCLUSION AND FUTURE WORKS

In this thesis, we proposed to enrich Wikipedia with multiple levels of annotation using its own structure. We developed an easy first iterative method to maximize the number of extracted annotations while introducing a reasonable amount of noise. We significantly extended the number of annotations of non anchored strings, thanks to coreference information and an analysis of the Wikipedia’s link structure. We applied our approach to a dump of English Wikipedia from 2013 and produced coarse- and fine-grained named-entity annotations. WiNER and WiFiNE surpasses other similar corpora, both in terms of quantity and of annotation quality. We evaluated annotation quality intrinsically (on manually labeled mentions) and extrinsically by using the corpus as training data for named-entity recognition, and fine-grained entity typing. We ran several experiments in order to use this massive amount of data in two ways:

1. directly as training material for entity typing supervised classifier.
2. indirectly as a source to generate classic and contextual word representations that can be easily added to supervised classifiers.

Our first endeavor through generating representations via supervised tasks consists in encoding classical lexicon features (word n -grams and gazetteer) into a compact form we called LS (**L**exical **S**imilarity). We represent each word by a 120-dimensional vector, where each dimension encodes the similarity of the word with an entity type.

Motivated by the great success and the gain in performances of ELMo [120] and BERT [41], we explored the utility of WiFiNE for learning contextualized word representations. Our main contribution lies in using semi-supervised approach for contextualized word representation learning. We trained an encoder that predicts the entity type of each input token. That is, compared to language modeling our model predicts an entity type rather than a word. Contrary to LS, the contextualized representation is a function of the entire sequence that can be a document.

We included our representations (LS and contextualized) as additional features in a vanilla LSTM-CRF NER model. The evaluation we conducted on 7 NER datasets of different genre and sizes shows that simply stacking various representations significantly boost performances across all domains over strong baselines. We reported systematic gains using the contextualized representation compared to models trained with LS. Ablation results show that our representations complement existing static and contextualized embeddings. Finally, we obtained new state-of-the-art performance¹ on two standard benchmarks: CONLL-2003 and ONTONOTES 5.0.

8.1 Future Work

At the time of writing this thesis, we observed an increasing interest in using Wikipedia in a distantly supervised setting. *DocRED* [172] is a newly created dataset of Wikipedia articles that are annotated with named-entities and relations for a document-level relation extraction task. The corpus has a manually labeled² version that consists of 5k documents with over 123k entities. Also, the authors produced a distantly supervised version of *DocRED* obtained by running the SpaCy [64] NER toolkit on 107k Wikipedia articles, which leads to a corpus with 2.2 millions annotated entities.

Abhishek et al. [3] revisit heuristic approaches to automatically convert Wikipedia to a high recall dataset for the fine-grained entity typing. The authors propose a staged pipeline which maps Wikipedia links to entities, matching them to Freebase, and then expanding them using some string matching, and then finally pruning sentences using heuristics. They evaluated their method by training state-of-the-art models trained on their new corpus. They show gains in recall and a small loss in precision when compared to the original existing wiki datasets with distant supervision. Their *Wiki-FbF* corpus is the most similar to WiFiNE, with two main differences. Because we match coreference mentions, the number of entities in *Wiki-FbF* is 4 times smaller than WiFiNE. While in WiFiNE we have rules that filter heterogeneous entity types (Section 4.4), this step

1. We report 93.2 and 89.9 on CONLL-2003 and ONTONOTES 5.0 respectively. The current state-of-the-art performances at the time of writing are 93.5 and 90.3 for the aforementioned datasets.

2. CONLL-2003 4 classes

is omitted by Abhishek et al. [3]. Although, the authors report higher performances on FIGER (GOLD), the results are not directly comparable because different models are used for training.

Zhu et al. [178] propose a two stage semi-supervised annotation framework to produce entity type annotations from Wikipedia abstracts³. In stage one, *AnchorNER*, is build by mapping Wikipedia hyperlinks to their respective types via attributes in DBpedia [9]. In stage two, the authors propose a machine learning approach, rather than heuristics, to rectify noisy labels. To improve the quality of the annotations, they leverage abstracts that exists in both *AnchorNER* and the manually-labeled *DocRED* dataset, where the latter labels are treated as ground truth. That is, the model is trained on words with their respective *AnchorNER* labels at the input, while the output are *DocRED* [172] gold labels. This model is applied on the rest of *AnchorNER* to correct the false-negative entity labels. The authors show systematic gains on 6 datasets when *AnchorNER* is used as training data. *AnchorNER* is similar to WiNER, nevertheless the authors did not perform a direct comparison with WiNER.

These approaches show that distant supervision with Wikipedia is a promising technique, however, they also reflect many limitations that must be addressed.

One major drawback of these approaches is the necessity of manually-labeled datasets in the annotation pipeline. For example, both WiNER and *AnchorNER* require Wikipedia manual annotated data. Further, the former corpus was build thanks to coreference annotations from *Wikicoref* [52], while the latter employs *DocRED* [172] gold labels to train a correction model. This limits the utility of these approaches for other languages even if Wikipedia exists in 282 languages. Future research must focus on refining automatic annotations without the need of human labeled data.

A big caveat of these approaches (including ours) is that it is very tailored towards Wikipedia texts, and it is unclear how it can generalize beyond. Since these approaches are meant for downstream training of NER models, they will be limited to Wikipedia-style texts. Or maybe it is not a limitation at all? It would be already good if we could show that NER systems trained on this Wikipedia-style texts do perform well in the open

3. That is, the first paragraph of a Wikipedia article

domain scenario, or under other challenging settings.

The extrinsic analysis of distantly supervised corpora, and measuring the robustness of NER models remains an open problem. In Section 5.2.5, we proposed OD_{F1} to measure the generalization performance on 6 datasets of different domains. Very recently, Lin et al. [89] conducted an empirical analysis on the robustness of NER models in the open domain scenario, and show that NER models are biased by strong name regularities, and the high mention coverage in standards benchmarks. Bernier-Colborne and Langlais [17] proposed `hardEval`, an evaluation protocol to measure the robustness of NER models on a subset of entities with unknown words, label shift, and those that are ambiguous.

Distant supervision assumptions have enabled the creation of large-scale datasets, the increased coverage often comes with increased noise. It could be researched how the quantity\quality trade-off in DS impacts the final performance in NER. For instance, *AnchorNER* [178] is relatively small compared to WiNER. The first is constructed from Wikipedia abstracts, while latter uses full articles. As noted by Nothman et al. [112], abstracts contain a high density of hyperlinks, but mostly full name of the entity is used as anchor. Abstract annotations have higher quality, but smaller size and less challenging training data comparing to annotations extracted from the entire article.

Although our overall 4 stage method shows gains on NER end performance, individual stages have their own limitations. As future work, it important to do a step back and revisit this 4 stage pipeline process, in order to identify these limitations. This can be done by directly measuring the gain of each step of the pipeline for the end task such as in Zhu et al. [178]. Also, it would be interesting to extrinsically compare between various methods for DS (WiNER and [178], WiFiNE and [3]), where models, training data size and test sets are immutable. To be comparable, it would require to apply our annotation process on recent Wikipedia dumps, with an up-to-date knowledge base such as WikiData [161]. This step is expected to further increase the number of entities, and also our rules would become more accurate by having access to more hyperlinks and KB relations.

A legitimate continuation of representation learning with distant supervision is to

measure if our classic or contextualized embeddings can improve results for other tasks by simply plugging them in existing models. We believe that our embeddings can improve results for tasks where entity type is crucial such as coreference resolution [83], relation extraction [106], and information extraction [149]. Preliminary experiments we conducted showed minor improvements for tasks like POS tagging, or sentiment analysis where entity type is not a key information for solving the task. We plan to study Relation Extraction (RE) because we believe that NE annotations to be useful.

Relation Extraction (RE) is defined as the task of extracting semantic relations between arguments. Arguments can either be entity types such as *Person*, *Organization*; or instances of such types (e.g. *Chilly Gonzales*, *Warner Bros*). Relation between entities are extracted from a predefined relation set such as `bornIn (PER, LOC)` and `founderOf (PER, ORG)`.

<p>Example (a): [<i>Hercule Poirot</i>]_{E1} is a fictional Belgian detective , created by [<i>Agatha Christie</i>]_{E2}.</p> <p>Relation: <code>character_created_by (E1, E2)</code></p> <hr/> <p>Example (b): [<i>Conan Doyle</i>]_{E1} acknowledged basing his detective stories on the model of [<i>Edgar Allan Poe</i>]_{E2} 's [<i>C. Auguste Dupin</i>]_{E3}.</p> <p>Relation: <code>influence_by (E1, E2) character_created_by (E3, E2)</code></p> <hr/> <p>Example (c): [<i>Gonzales</i>]_{E1} said in an interview: « My experiences in [<i>Canada</i>]_{E2} had been disappointing. »</p> <p>Relation: <code>nationality (E1, E2)</code></p> <hr/> <p>Example (d): When Christie 's daughter , Rosalind Hicks , observed Ustinov during a rehearsal , [<i>she</i>]_{E1} said: « That 's not [<i>Poirot</i>]_{E2} ! »</p> <p>Relation: <code>character_created_by (E2, E1)</code></p>

Figure 8.1 – Example of correct (a-b) and wrong (c-d) relations annotated using distant supervision.

The task is challenging because relations may not be explicitly expressed in the sentence. It is a pipelined classification task that consists of two phases: named-entity

identification and relation classification. One way to apply distant supervision for this task would be to assume that a relation in a sentence holds between 2 arguments in a sentence hold if both arguments are present in a triple in Freebase. Figure 8.1 shows examples of correct (example *a* and *b*), and noisy (example *c* and *d*) triples obtained on our Wikipedia enriched corpus.

BIBLIOGRAPHY

- [1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL <http://tensorflow.org/>. Software available from tensorflow.org.
- [2] Abhishek Abhishek, Ashish Anand, and Amit Awekar. Fine-Grained Entity Type Classification by Jointly Learning Representations and Label Embeddings. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 797–807, Valencia, Spain, April 2017. Association for Computational Linguistics.
- [3] Abhishek Abhishek, Sanya Bathla Taneja, Garima Malik, Ashish Anand, and Amit Awekar. Fine-grained entity recognition with reduced false negatives and large type coverage. *arXiv preprint arXiv:1904.13178*, 2019.
- [4] Alan Akbik, Duncan Blythe, and Roland Vollgraf. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649, 2018.
- [5] Rami Al-Rfou, Vivek Kulkarni, Bryan Perozzi, and Steven Skiena. Polyglot-ner: Massive multilingual named entity recognition. In *Proceedings of the 2015 SIAM International Conference on Data Mining, Vancouver, British Columbia, Canada*. SIAM, 2015.
- [6] Julio Cesar Salinas Alvarado, Karin Verspoor, and Timothy Baldwin. Domain

- adaption of named entity recognition to support credit risk assessment. In *Proceedings of the Australasian Language Technology Association Workshop 2015*, pages 84–90, 2015.
- [7] Daniel Andor, Luheng He, Kenton Lee, and Emily Pitler. Giving BERT a Calculator: Finding Operations and Arguments with Reading Comprehension. 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 5949–5954, 2019.
- [8] Giuseppe Attardi. Deepnl: a deep learning nlp pipeline. In *Proceedings of Workshop on Vector Space Modeling for NLP, NAACL-HLT*, pages 109–115, 2015.
- [9] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. In *The Semantic Web*, pages 722–735. Springer, 2007.
- [10] Isabelle Augenstein, Leon Derczynski, and Kalina Bontcheva. Generalisation in named entity recognition: A quantitative analysis. *Computer Speech & Language*, 44:61–83, 2017.
- [11] Alexei Baevski, Sergey Edunov, Yinhan Liu, Luke Zettlemoyer, and Michael Auli. Cloze-driven Pretraining of Self-attention Networks. 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 5363–5372, 2019.
- [12] Dzmitry Bahdanau, Tom Bosc, Stanisław Jastrzębski, Edward Grefenstette, Pascal Vincent, and Yoshua Bengio. Learning to compute word embeddings on the fly. *arXiv preprint arXiv:1706.00286*, 2017.
- [13] Dominic Balasuriya, Nicky Ringland, Joel Nothman, Tara Murphy, and James R Curran. Named entity recognition in wikipedia. In *Proceedings of the 2009 Work-*

- shop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources*, pages 10–18. Association for Computational Linguistics, 2009.
- [14] Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. What do Neural Machine Translation Models Learn about Morphology? In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 861–872, 2017.
- [15] Richard Bellman. Dynamic programming. *Science*, 153(3731):34–37, 1966.
- [16] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb): 1137–1155, 2003.
- [17] Gabriel Bernier-Colborne and Phillippe Langlais. HardEval: Focusing on Challenging Tokens to Assess Robustness of NER. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 1697–1704, Marseille, France, May 2020. European Language Resources Association.
- [18] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.
- [19] 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.
- [20] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, pages 1247–1250, 2008.
- [21] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, pages 1247–1250, 2008.

- [22] Samuel Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, 2015.
- [23] Peter F Brown, Peter V Desouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai. Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–479, 1992.
- [24] Ming-Wei Chang, Kristina Toutanova, Kenton Lee, and Jacob Devlin. Language Model Pre-training for Hierarchical Document Representations. *arXiv preprint arXiv:1901.09128*, 2019.
- [25] Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. One Billion Word Benchmark for Measuring Progress in Statistical Language Modeling. In *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.
- [26] Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. Enhanced LSTM for Natural Language Inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1657–1668, 2017.
- [27] Xue-Wen Chen and Xiaotong Lin. Big data deep learning: challenges and perspectives. *IEEE access*, 2:514–525, 2014.
- [28] Nancy Chinchor and Beth Sundheim. Message understanding conference (muc) 6. *LDC2003T13*, 2003.
- [29] Nancy A Chinchor. Overview of muc-7/met-2. Technical report, SCIENCE APPLICATIONS INTERNATIONAL CORP SAN DIEGO CA, 1998.
- [30] Jason PC Chiu and Eric Nichols. Named entity recognition with bidirectional lstm-cnns. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, 2016.

- [31] Kevin Clark, Minh-Thang Luong, Christopher D Manning, and Quoc Le. Semi-Supervised Sequence Modeling with Cross-View Training. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1914–1925, 2018.
- [32] 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.
- [33] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537, 2011.
- [34] Donald C Comeau, Rezarta Islamaj Doğan, Paolo Ciccarese, Kevin Bretonnel Cohen, Martin Krallinger, Florian Leitner, Zhiyong Lu, Yifan Peng, Fabio Rinaldi, Manabu Torii, et al. Bioc: a minimalist approach to interoperability for biomedical text processing. *Database*, 2013, 2013.
- [35] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, 2017.
- [36] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.
- [37] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G Carbonell, Quoc Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive Language Models beyond a Fixed-Length Context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, 2019.
- [38] Joachim Daiber, Max Jakob, Chris Hokamp, and Pablo N. Mendes. Improving Efficiency and Accuracy in Multilingual Entity Extraction. In *Proceedings of the 9th International Conference on Semantic Systems (I-Semantics)*, 2013.

- [39] Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 933–941. JMLR. org, 2017.
- [40] Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. Results of the wnut2017 shared task on novel and emerging entity recognition. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 140–147, 2017.
- [41] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.
- [42] George R. Doddington, Alexis Mitchell, Mark A. Przybocki, Lance A. Ramshaw, Stephanie Strassel, and Ralph M. Weischedel. The Automatic Content Extraction (ACE) Program-Tasks, Data, and Evaluation. In *LREC*, volume 2, page 1, 2004.
- [43] Samuel Dodge and Lina Karam. Understanding how image quality affects deep neural networks. In *Quality of Multimedia Experience (QoMEX), 2016 Eighth International Conference on*, pages 1–6. IEEE, 2016.
- [44] Rezarta Islamaj Doğan, Robert Leaman, and Zhiyong Lu. NCBI disease corpus: a resource for disease name recognition and concept normalization. *Journal of biomedical informatics*, 47:1–10, 2014.
- [45] 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.
- [46] Fredo Erxleben, Michael Günther, Markus Krötzsch, Julian Mendez, and Denny Vrandečić. Introducing Wikidata to the linked data web. In *International Semantic Web Conference*, pages 50–65. Springer, 2014.

- [47] Michael Ringgaard Evgeniy Gabrilovich and Amarnag Subramanya. Facc1: Free-base annotation of clueweb corpora, version 1 (release date 2013-06-26, format version 1, correction level 0). June 2013.
- [48] Paolo Ferragina and Ugo Scaiella. Tagme: on-the-fly annotation of short text fragments (by wikipedia entities). In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1625–1628. ACM, 2010.
- [49] Jenny Rose Finkel and Christopher D Manning. Joint Parsing and Named Entity Recognition. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 326–334. Association for Computational Linguistics, 2009.
- [50] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370. Association for Computational Linguistics, 2005.
- [51] Abbas Ghaddar. Coreference Resolution with and for Wikipedia. Masters, Université de Montréal, Montréal, 09 2016.
- [52] Abbas Ghaddar and Philippe Langlais. WikiCoref: An English Coreference-annotated Corpus of Wikipedia Articles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Portorož, Slovenia, 05/2016 2016.
- [53] 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.
- [54] Abbas Ghaddar and Philippe Langlais. Contextualized Word Representations from Distant Supervision with and for NER. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 101–108, 2019.

- [55] Abbas Ghaddar and Phillippe Langlais. Coreference in Wikipedia: Main concept resolution. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 229–238, 2016.
- [56] Abbas Ghaddar and Phillippe Langlais. Winer: A wikipedia annotated corpus for named entity recognition. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 413–422, 2017.
- [57] Abbas Ghaddar and Phillippe Langlais. Robust Lexical Features for Improved Neural Network Named-Entity Recognition. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1896–1907, 2018.
- [58] Dan Gillick, Nevena Lazic, Kuzman Ganchev, Jesse Kirchner, and David Huynh. Context-dependent fine-grained entity type tagging. *arXiv preprint arXiv:1412.1820*, 2014.
- [59] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.
- [60] Ralph Grishman and Beth Sundheim. Message understanding conference-6: A brief history. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*, 1996.
- [61] Luheng He, Kenton Lee, Omer Levy, and Luke Zettlemoyer. Jointly Predicting Predicates and Arguments in Neural Semantic Role Labeling. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 364–369, 2018.
- [62] Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. Scalable modified Kneser-Ney language model estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 690–696, Sofia, Bulgaria, August 2013.

- [63] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [64] Matthew Honnibal and Ines Montani. spacy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. *To appear*, 7(1), 2017.
- [65] Matthew Honnibal and Ines Montani. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear, 2017.
- [66] Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. OntoNotes: the 90% solution. In *Proceedings of the human language technology conference of the NAACL, Companion Volume: Short Papers*, pages 57–60. Association for Computational Linguistics, 2006.
- [67] Jeremy Howard and Sebastian Ruder. Universal Language Model Fine-tuning for Text Classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, 2018.
- [68] Luyao Huang, Chi Sun, Xipeng Qiu, and Xuan-Jing Huang. GlossBERT: BERT for Word Sense Disambiguation with Gloss Knowledge. 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 3500–3505, 2019.
- [69] Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*, 2015.
- [70] 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.

- [71] Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410*, 2016.
- [72] Gjergji Kasneci, Maya Ramanath, Fabian Suchanek, and Gerhard Weikum. The yago-naga approach to knowledge discovery. *ACM SIGMOD Record*, 37(4):41–47, 2009.
- [73] J-D Kim, Tomoko Ohta, Yuka Tateisi, and Jun’ichi Tsujii. GENIA corpus—a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(suppl_1):i180–i182, 2003.
- [74] Yoon Kim. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, 2014.
- [75] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [76] Vijay Krishnan and Christopher D Manning. An effective two-stage model for exploiting non-local dependencies in named entity recognition. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 1121–1128. Association for Computational Linguistics, 2006.
- [77] Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. RACE: Large-scale ReAding Comprehension Dataset From Examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, 2017.
- [78] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural Architectures for Named Entity Recognition. In *Proceedings of NAACL-HLT*, pages 260–270, 2016.

- [79] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.
- [80] 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, 2019.
- [81] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521 (7553):436, 2015.
- [82] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 2020.
- [83] 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, 2017.
- [84] 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, 2018.
- [85] Omer Levy and Yoav Goldberg. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 302–308, 2014.
- [86] Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. A survey on deep learning for named entity recognition. *arXiv preprint arXiv:1812.09449*, 2018.
- [87] Yaoyong Li, Kalina Bontcheva, and Hamish Cunningham. SVM based learning system for information extraction. In *International Workshop on Deterministic and Statistical Methods in Machine Learning*, pages 319–339. Springer, 2004.

- [88] Dekang Lin and Xiaoyun Wu. Phrase Clustering for Discriminative Learning. 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: Volume 2-Volume 2*, pages 1030–1038. Association for Computational Linguistics, 2009.
- [89] Hongyu Lin, Yaojie Lu, Xianpei Han, and Le Sun. A Rigorous Study on Named Entity Recognition: Can Fine-tuning Pretrained Model Lead to the Promised Land? *arXiv preprint arXiv:2004.12126*, 2020.
- [90] Wang Ling, Yulia Tsvetkov, Silvio Amir, Ramon Fernandez, Chris Dyer, Alan W Black, Isabel Trancoso, and Chu-Cheng Lin. Not all contexts are created equal: Better word representations with variable attention. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1367–1372, 2015.
- [91] Xiao Ling and Daniel S Weld. Fine-grained entity recognition. In *AAAI*, 2012.
- [92] Xiao Ling, Sameer Singh, and Daniel S Weld. Design challenges for entity linking. *Transactions of the Association for Computational Linguistics*, 3:315–328, 2015.
- [93] Liyuan Liu, Jingbo Shang, Xiang Ren, Frank Fangzheng Xu, Huan Gui, Jian Peng, and Jiawei Han. Empower sequence labeling with task-aware neural language model. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- [94] Xiaodong Liu, Yelong Shen, Kevin Duh, and Jianfeng Gao. Stochastic Answer Networks for Machine Reading Comprehension. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1694–1704, 2018.
- [95] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A

- robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.
- [96] Peng Lu, Ting Bai, and Philippe Langlais. SC-LSTM: Learning Task-Specific Representations in Multi-Task Learning for Sequence Labeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2396–2406, 2019.
- [97] Gang Luo, Xiaojiang Huang, Chin-Yew Lin, and Zaiqing Nie. Joint Entity Recognition and Disambiguation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 879–888, Lisbon, Portugal, September 2015. Association for Computational Linguistics.
- [98] Xuezhe Ma and Eduard Hovy. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, 2016.
- [99] Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*, pages 6297–6308, 2017.
- [100] Tom McCoy, Ellie Pavlick, and Tal Linzen. Right for the Wrong Reasons: Diagnosing Syntactic Heuristics in Natural Language Inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, 2019.
- [101] Olena Medelyan, David Milne, Catherine Legg, and Ian H. Witten. Mining meaning from wikipedia. *Int. J. Hum.-Comput. Stud.*, 67(9):716–754, September 2009.
- [102] Tomas Mikolov, Jiri Kopecky, Lukas Burget, Ondrej Glembek, et al. Neural network based language models for highly inflective languages. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4725–4728. IEEE, 2009.

- [103] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
- [104] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.
- [105] Tomáš Mikolov, Édouard Grave, Piotr Bojanowski, Christian Puhersch, and Armand Joulin. Advances in Pre-Training Distributed Word Representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.
- [106] 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: Volume 2-Volume 2*, pages 1003–1011. Association for Computational Linguistics, 2009.
- [107] Roberto Navigli. Word sense disambiguation: A survey. *ACM computing surveys (CSUR)*, 41(2):10, 2009.
- [108] Arvind Neelakantan, Jeevan Shankar, Alexandre Passos, and Andrew McCallum. Efficient Non-parametric Estimation of Multiple Embeddings per Word in Vector Space. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1059–1069, 2014.
- [109] Rodrigo Nogueira and Kyunghyun Cho. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*, 2019.
- [110] Thanapon Noraset, Chandra Bhagavatula, and Doug Downey. Adding high-precision links to wikipedia. In *EMNLP*, pages 651–656, 2014.
- [111] Joel Nothman. *Learning named entity recognition from Wikipedia*. PhD thesis, The University of Sydney Australia 7, 2008.

- [112] Joel Nothman, James R Curran, and Tara Murphy. Transforming wikipedia into named entity training data. In *Proceedings of the Australian Language Technology Workshop*, pages 124–132, 2008.
- [113] Joel Nothman, Tara Murphy, and James R Curran. Analysing Wikipedia and gold-standard corpora for NER training. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 612–620. Association for Computational Linguistics, 2009.
- [114] Joel Nothman, Nicky Ringland, Will Radford, Tara Murphy, and James R Curran. Learning multilingual named entity recognition from wikipedia. *Artificial Intelligence*, 194:151–175, 2013.
- [115] Alexandre Passos, Vineet Kumar, and Andrew McCallum. Lexicon Infused Phrase Embeddings for Named Entity Resolution. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 78–86, 2014.
- [116] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [117] Nanyun Peng, Hoifung Poon, Chris Quirk, Kristina Toutanova, and Wen-tau Yih. Cross-sentence n-ary relation extraction with graph lstms. *Transactions of the Association for Computational Linguistics*, 5:101–115, 2017.
- [118] 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, 2014.
- [119] 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, 2017.

- [120] 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, 2018.
- [121] Matthew Peters, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih. Dissecting Contextual Word Embeddings: Architecture and Representation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1499–1509, 2018.
- [122] Marten Postma, Filip Ilievski, Piek Vossen, and Marieke van Erp. Moving away from semantic overfitting in disambiguation datasets. *EMNLP 2016*, page 17, 2016.
- [123] 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, 2012.
- [124] Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. Towards robust linguistic analysis using ontonotes. In *CoNLL*, pages 143–152, 2013.
- [125] Sameer S. Pradhan, Lance Ramshaw, Ralph Weischedel, Jessica MacBride, and Linnea Micciulla. Unrestricted coreference: Identifying entities and events in OntoNotes. In *First IEEE International Conference on Semantic Computing*, pages 446–453, 2007.
- [126] Chris Quirk and Hoifung Poon. Distant supervision for relation extraction beyond the sentence boundary. In *Proceedings of the 15th Conference of the European*

Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, pages 1171–1182, 2017.

- [127] 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/researchcovers/languageunsupervised/language_understanding_paper.pdf, 2018.
- [128] Alessandro Raganato, Claudio Delli Bovi, and Roberto Navigli. Automatic construction and evaluation of a large semantically enriched wikipedia. In *IJCAI*, pages 2894–2900, 2016.
- [129] 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, 2010.
- [130] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, 2016.
- [131] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know What You Don’t Know: Unanswerable Questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, 2018.
- [132] Lev Ratinov and Dan Roth. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pages 147–155. Association for Computational Linguistics, 2009.
- [133] Siva Reddy, Mirella Lapata, and Mark Steedman. Large-scale semantic parsing

- without question-answer pairs. *Transactions of the Association for Computational Linguistics*, 2:377–392, 2014.
- [134] Tom Redman, Mark Sammons, and Dan Roth. Illinois Named Entity Recognizer: Addendum to Ratinov and Roth '09 reporting improved results, 2016.
- [135] 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 *EMNLP*, volume 16, page 17, 2016.
- [136] 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.
- [137] Alexander E. Richman and Patrick Schone. Mining wiki resources for multilingual named entity recognition in proceedings. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, pages 1–9, 2004.
- [138] Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. Named entity recognition in tweets: An experimental study. In *EMNLP*, 2011.
- [139] Michael Röder, Ricardo Usbeck, Sebastian Hellmann, Daniel Gerber, and Andreas Both. N³-A Collection of Datasets for Named Entity Recognition and Disambiguation in the NLP Interchange Format. In *LREC*, pages 3529–3533, 2014.
- [140] Evan Sandhaus. The new york times annotated corpus. *Linguistic Data Consortium, Philadelphia*, 6(12):e26752, 2008.
- [141] Saskia Schön, Veselina Mironova, Aleksandra Gabryszak, and Leonhard Hennig. A corpus study and annotation schema for named entity recognition and relation extraction of business products. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, 2018.

- [142] Yanyao Shen, Hyokun Yun, Zachary C Lipton, Yakov Kronrod, and Animashree Anandkumar. Deep Active Learning for Named Entity Recognition. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 252–256, 2017.
- [143] 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.
- [144] Avirup Sil, Gourab Kundu, Radu Florian, and Wael Hamza. Neural cross-lingual entity linking. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- [145] Sameer Singh, Amarnag Subramanya, Fernando Pereira, and Andrew McCallum. Wikilinks: A large-scale cross-document coreference corpus labeled via links to wikipedia. *University of Massachusetts, Amherst, Tech. Rep. UM-CS-2012-015*, 2012.
- [146] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.
- [147] Anders Søgaard and Yoav Goldberg. Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, volume 2, pages 231–235, 2016.
- [148] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15(1):1929–1958, 2014.
- [149] Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. Supervised open information extraction. 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 885–895, 2018.

- [150] Benjamin Strauss, Bethany Toma, Alan Ritter, Marie-Catherine de Marneffe, and Wei Xu. Results of the wnut16 named entity recognition shared task. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 138–144, 2016.
- [151] Emma Strubell, Patrick Verga, David Belanger, and Andrew McCallum. Fast and Accurate Entity Recognition with Iterated Dilated Convolutions. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2660–2670, 2017.
- [152] Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. Linguistically-Informed Self-Attention for Semantic Role Labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5027–5038, 2018.
- [153] Amber Stubbs and Özlem Uzuner. Annotating longitudinal clinical narratives for de-identification: The 2014 i2b2/UTHealth corpus. *Journal of biomedical informatics*, 58:S20–S29, 2015.
- [154] 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-Volume 4*, pages 142–147. Association for Computational Linguistics, 2003.
- [155] A. Toral and R. Munoz. A proposal to automatically build and maintain gazetteers for Named Entity Recognition by using Wikipedia. In *Proceedings of the EACL-2006 Workshop on New Text: Wikis and blogs and other dynamic text sources-EACL Workshop on NEW TEXT-Wikis and blogs and ther dynamic text sources*, pages 56–61, 2006.

- [156] Quan Hung Tran, Andrew MacKinlay, and Antonio Jimeno Yepes. Named Entity Recognition with Stack Residual LSTM and Trainable Bias Decoding. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 566–575, 2017.
- [157] Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 384–394. Association for Computational Linguistics, 2010.
- [158] L.J.P. van der Maaten. Accelerating t-SNE using Tree-Based Algorithms. *Journal of Machine Learning Research*, 15:3221–3245, 2014.
- [159] Marieke van Erp, Pablo N Mendes, Heiko Paulheim, Filip Ilievski, Julien Plu, Giuseppe Rizzo, and Jörg Waitelonis. Evaluating Entity Linking: An Analysis of Current Benchmark Datasets and a Roadmap for Doing a Better Job. In *LREC*, volume 5, page 2016, 2016.
- [160] 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.
- [161] Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledge base. 2014.
- [162] Ralph Weischedel and Ada Brunstein. Bbn pronoun coreference and entity type corpus. *Linguistic Data Consortium, Philadelphia*, 112, 2005.
- [163] Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, et al. Ontonotes release 5.0 ldc2013t19. *Linguistic Data Consortium, Philadelphia, PA*, 2013.
- [164] Robert West, Ashwin Paranjape, and Jure Leskovec. Mining missing hyperlinks from human navigation traces: A case study of wikipedia. In *Proceedings of*

the 24th international conference on World Wide Web, pages 1242–1252. International World Wide Web Conferences Steering Committee, 2015.

- [165] Michael Wick, Sameer Singh, Harshal Pandya, and Andrew McCallum. A joint model for discovering and linking entities. In *Proceedings of the 2013 workshop on Automated knowledge base construction*, pages 67–72. ACM, 2013.
- [166] Adina Williams, Nikita Nangia, and Samuel R Bowman. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*, 2017.
- [167] 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.
- [168] Jie Yang, Yue Zhang, and Fei Dong. Neural Reranking for Named Entity Recognition. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 784–792, 2017.
- [169] Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. End-to-End Open-Domain Question Answering with BERT-serini. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 72–77, 2019.
- [170] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764, 2019.
- [171] Xuchen Yao and Benjamin Van Durme. Information extraction over structured data: Question answering with freebase. In *ACL (1)*, pages 956–966, 2014.

- [172] Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. DocRED: A Large-Scale Document-Level Relation Extraction Dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 764–777, 2019.
- [173] Dani Yogatama, Daniel Gillick, and Nevena Lazic. Embedding Methods for Fine Grained Entity Type Classification. In *ACL (2)*, pages 291–296, 2015.
- [174] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.
- [175] Wang Ling Lin Chu-Cheng Yulia, Tsvetkov Silvio Amir, Ramón Fernandez Astudillo Chris Dyer Alan, and W Black Isabel Trancoso. Not all contexts are created equal: Better word representations with variable attention. 2015.
- [176] Matthew D Zeiler. ADADELTA: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.
- [177] Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. SWAG: A Large-Scale Adversarial Dataset for Grounded Commonsense Inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 93–104, 2018.
- [178] Mengdi Zhu, Zheyue Deng, Wenhan Xiong, Mo Yu, Ming Zhang, and William Yang Wang. Towards open-domain named entity recognition via neural correction models. *arXiv preprint arXiv:1909.06058*, 2019.