The Utility and Interplay of Gazetteers and Entity Segmentation for Named Entity Recognition in English

Oshin Agarwal[1] and **Ani Nenkova**[1,2]
[1]University of Pennsylvania
[2]Adobe Research
oagarwal@seas.upenn.edu, nenkova@adobe.com

Abstract

Recent papers have introduced methods to incorporate gazetteer features and entity segmentation techniques in neural named entity recognition models. These papers rely on different resources and include features not related to the use of gazetteers, rendering impossible the comparison of the relative effectiveness of the approaches. Here, we provide a comprehensive overview of methods for incorporating gazetteers and for entity segmentation. We evaluate representative methods from each in similar settings for a fair comparison and identify the ones that are consistently better across datasets and input representations. We further show that gazetteers improve entity segmentation and not just entity typing. Hence, we explore their utility in recognizing long entities, a problem for which entity segmentation techniques were developed. Our work explains the mechanisms via which gazetteers improve the performance of neural NER models.

1 Introduction

Named Entity Recognition (NER) has the unique property of being a task appealing to researchers and at the same time being fairly robust for immediate practical applications. In many domains, it is of interest to identify segments of text conveying a concept of a given type—a person (Grishman and Sundheim, 1996), an event (Hovy et al., 2006), a disease (Doğan et al., 2014), a gene (Kim et al., 2003), a chemical (Krallinger et al., 2015), a food (Magnolini et al., 2019), an item of clothing (Putthividhya and Hu, 2011), a research technique (Augenstein et al., 2017a), etc.

Approaches to NER are typically not domain-specific, treating the problem as a sequence labelling task regardless of the categories of interest. Yet, researchers also widely agree that named entity recognition is a knowledge intensive task (Ratinov and Roth, 2009; Seyler et al., 2018): the availability of external knowledge resources in the form of lists of example entities of a given type, or *gazetteers*, improve performance almost universally. Since gazetteers are readily available, from knowledge bases, databases of products and specialized ontologies, having practical guidance on how to handle gazetteers in NER would be valuable.

In this paper, we provide a survey of how gazetteers have been used in neural approaches to NER in English and compare key approaches with the popular biLSTM-CRF architecture. To ensure that our conclusions accurately characterize the utility of gazetteers, we test the approaches on several datasets from different genres covering newswire, conversations and twitter. The extensive head-to-head comparison reveals that while certain approaches are consistently beneficial, others are variable, impressively improving results on one dataset but reducing performance on others.

Gazetteers typically contain multi-word entries. In contrast, the majority of entity mentions in text are single word, with lower performance of models on longer entities. This discrepancy highlights a potential application of gazetteers for improving NER prediction for mentions of long entities. Recent work on entity segmentation[1] as part of the named entity recognition task aims to recognize long entities better. We overview this work and compare these methods similar to the way we compare methods for incorporating gazetteers.

We find that certain ways for incorporating gazetteers show stable improvements across datasets, while segmentation approaches do not appear to be that useful overall. We further explore the interplay of gazetteers with entity segmentation and their role in recognizing long entities. We find that incorporating gazetteers improves entity

[1]Identifying entity spans without types

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 3990–4002

segmentation, not just entity typing and depending on the input representation to the model, gazetteer types may be irrelevant. We also find that incorporating gazetteers can serve as an alternate method to recognizing long entities, likely due to the abundant presence of multi-word entities in the Wikipedia-derived gazetteers we used.

Our work provides *(i)* a concise overview of methods for incorporating gazetteers and entity segmentation in NER, *(ii)* a principled comparison of representative approaches for each aspect, and *(iii)* novel findings and analyses of the interplay between gazetteers and segmentation. Our findings can inform both future researchers and practitioners interested in NER.

2 biLSTM-CRF architecture for NER

We explore variants of the now classic biLSTM-CRF architecture for NER (Huang et al., 2015). We overview how gazetteer features and segmentation can be integrated in this paradigm and carry out a comparison of several representative methods. We use two input word representations: the 300-d GloVe vectors trained on Common Crawl (Pennington et al., 2014), which is the dominant representation in NER, and the 1024-d contextual ELMo (Peters et al., 2018) representations trained on the 1B Word Benchmark (Chelba et al., 2014). In each case, character-based word representations learned with CNNs (Ma and Hovy, 2016) are also concatenated. The final concatenated representation is used as input to bidirectional LSTMs (Hochreiter and Schmidhuber, 1997), followed by a CRF (Lafferty et al., 2001) layer. We use the implementation by Lample et al. (2016) for most experiments.

3 Why Use Gazetteers?

Gazetteers are large dictionaries consisting of lists of entities of a particular type. For example, a person gazetteer may consist of full names and parts of names such as the first names of people. Before we start our discussion of methods for incorporating gazetteers, it is worth considering if we have sufficient evidence that they are needed at all.

Gazetteers are needed for better generalization through improved entity coverage, to predict the type for words that have not been encountered in training and possibly even pre-training. This means the need will be more acute for practical deployment of NER and will be less pronounced on fixed datasets in which train and test data are sampled from overlapping or adjacent time periods, with high overlap of entities across both.

The most compelling example for the need to handle unseen language comes from work on NER on Twitter. Language on Twitter changes rapidly, much more rapidly than for other types of text (Eisenstein, 2013). This change requires models to be retrained periodically to maintain optimal performance for the current time period (Rijhwani and Preotiuc-Pietro, 2020). An alternative to retraining, not yet explored in literature, is to develop methods that can make use of gazetteers that possibly could be updated more quickly and cheaply compared to continuously annotating new training data.

Even in stable domains such as newswire, the ability of models to generalize to words not seen in the training data is low (Augenstein et al., 2017b; Fu et al., 2020a,b). Both traditional models with hand-crafted features (Finkel et al., 2005; Okazaki, 2007) and more recent neural network approaches (Collobert et al., 2011; Huang et al., 2015; Peters et al., 2018; Devlin et al., 2019) achieve lower performance on entities unseen in the training data.

Methods that make use of large pretrained language representations, neural (Collobert et al., 2011) or not (Miller et al., 2004), can ameliorate the problem of coverage to some extent. We do not yet know enough about how pretraining data should be chosen (Cherry and Guo, 2015), though there is some evidence that performance on downstream tasks correlates with the vocabulary coverage in the pre-training data (Dai et al., 2019). Prior work has reported that performance is lowest for words that appear neither in the training nor the pretraining vocabulary (Ma and Hovy, 2016). Moreover, the deteriorated performance on out of vocabulary words is not necessarily a failure of the models: many contexts simply do not provide sufficient knowledge to predict the type of an entity, even for people (Agarwal et al., 2021). Models need to expand their knowledge of entities and gazetteers are a natural way for doing that.

4 Where Do Gazetteers Come From?

Existing tables, lists, directories, databases and knowledge bases are widely available and can be used to derive gazetteers. Some researchers have specifically compiled various resources to form gazetteers, while others make use of those provided in prior work. Early work collected gazetteers from the CIA factbook for geographic locations, lists of

1	2	3	≥ 4
15	43	18	24

Table 1: Gazetteer entities(%) by length (words).

popular person names, etc (Mikheev et al., 1999). More recently, Ratinov and Roth (2009) derived a gazetteer from the Web and Wikipedia and Chiu and Nichols (2016) used DBPedia.

In our work, we use the Ratinov and Roth (2009) wide-coverage gazetteer. It contains ∼3M entities grouped into ∼30 fine-grained categories. In some experiments, we use all categories, regardless of the entity types in the dataset. In the remaining, we identify the gazetteer category that most closely matches that types in a given dataset and disregard the rest. The mapping can be found in the appendix.

Table 1 shows the approximate percentage of entities by length in words in our gazetteer. Most entries are of length two, but unlike NER datasets (Table 3), the remaining entries are evenly distribution between length 1, 3 and ≥4. Most notably, around 24% of gazetteer entities have four or more words. In comparison, in NER datasets where entities appear in the context of a sentence that is often a part of a longer document, such long entities typically make up about 2% of all entities. This distributional difference hints at the possibility of using gazetteer to not only improve coverage but also improve performance on longer entities for which segmentation methods are developed.

Regardless of the distribution of entity lengths, the total number of entities in gazetteers is much higher than that in NER datasets so a higher percentage of longer entities does not equate to a small number of short entities.

5 Gazetteer Features for NER

Here, we overview the ways gazetteers have been integrated in NER models.

5.1 Discrete Gazetteer Lookup Features

Feature-based CRF models for NER used gazetteers to generate indicators for each word in a sentence (Bender et al., 2003; Minkov et al., 2005; Ratinov and Roth, 2009; Ritter et al., 2011; Yang et al., 2016; Seyler et al., 2018). The number of indicators equals the number of entity types in the dataset and indicate (with a binary 1/0 value) if the word is part of a gazetteer entry of the given type.

Many neural network approaches continue to incorporate gazetteers as discrete indicator features concatenated to the pre-trained word embeddings as the input (Collobert et al., 2011; Huang et al., 2015). Adding the features in later stages does not work as well (Magnolini et al., 2019). Both Collobert et al. (2011) and Huang et al. (2015) pre-process datasets to match the gazetteer entries to sentences, using both exact matches and multi-word partial matches to gazetteer entries. Chiu and Nichols (2016) perform a similar matching but use four binary values for each label, indicating whether the given word matches the gazetteer entity exactly (S), at the beginning (B), end (E) or the any of the words in between (I).

5.2 Continuous Gazetteer Features

The approach above does not use gazetteers very effectively. Gazetteers contain many more entities of each type than are available in even the largest training set (Table 3). One insight is to use the gazetteers as a additional source of training examples. A simple way is to add the gazetteer entries to the labeled data, without any context. Liu et al. (2019a) report that this data augmentation approach led to much worse overall results, presumably because of the great shift in label distributions. Another approach is to augment the training data by replacing entities in place by other entities from gazetteers. Song et al. (2020) reported no improvement with such a random entity replacement, likely due to the need for manual intervention for replacement of entities of some types to maintain coherence of text (Agarwal et al., 2020).

A much more successful alternative is to learn a separate (or sub-) module, trained to predict types for text spans, using the gazetteer entries and synthetic negative examples sampled from a NER training set or even the gazetteer. We will refer to the separate module as a *gazetteer network*. It is straightforward to integrate the label distribution scores from this model in a semi-Markov CRF for sequence labeling (Ye and Ling, 2018) that operates at the span level (which we describe in greater detail later). The resulting combination is far more effective than discrete indicator gazetteer features.

Magnolini et al. (2019) and Liu et al. (2019b) propose a similar approach. They learn a gazetteer network but instead of using the label score distribution, intermediate word representations (*gazetteer embeddings* henceforth) are incorporated in the NER model. Liu et al. (2019b) use a semi-Markov CRF operating at the span level and generate the

gazetteer embeddings for each potential span. They follow the evaluation approach of Ma and Hovy (2016), breaking down results by whether an entity was seen only in training, seen only in pre-training, seen in both and seen in neither. The largest improvement was in the "seen in neither" subset, showing that this approach is particularly helpful for out-of-vocabulary words with respect to the training and pre-training data.

Magnolini et al. (2019) use the standard word-level CRF and hence do not have spans available so they input the full sentence to the gazetteer network. This makes the training and inference setup for the gazetteer network different as entity phrases are used as input during training. They reported mixed results for this approach. In our experiments, we evaluated their method on a larger number of datasets but used a different approach for negative sampling for the gazetteer network training data. We observed some improvement on almost all datasets, contingent on the input representation.

5.3 Contextual Gazetteers

Learning from just the gazetteer has the drawback that the representations do not include any clues about the context in which the entity types are used. The same entity may appear in multiple gazetteers. Given that current methods heavily rely on entity memorization and little on context, this is possibly acceptable. For completeness however, we ought to mention that the link structure of Wikipedia can be used to derive dense representations for entity types directly (Long et al., 2016; Ganea and Hofmann, 2017; Mengge et al., 2020; Ghaddar and Langlais, 2018). Comparing gazetteer representations with and without context would be a direction for exploration in future work.

6 Entity Segmentation in NER

Early work (Collins and Singer, 1999; Downey et al., 2007; Ritter et al., 2011) treated NER as two subtasks, i.e. *entity segmentation:* finding spans of text that refer to named entities, and *typing:* assigning a type to the identified span. Recent efforts have also incorporated entity segmentation explicitly in neural models, with goal of finding longer entities better (Xiao et al., 2019; Ye and Ling, 2018). Such work can be divided into two categories—Multi-task learning and semi-Markov CRFs.

6.1 Multi-task Learning

Multi-task learning (MTL) involves jointly training multiple related tasks using the same representation such that the auxiliary tasks can help with the performance of the target task. Aguilar et al. (2017) use MTL with hard parameter sharing to add two auxiliary tasks—binary classification to identify entity spans (segmentation) and multi-class classification to type them without CRF. Both tasks use the same biLSTM representation as the target task. The output of the additional tasks is not used in the NER task; they only act as regularizers for NER. Others (Stratos, 2017; Aguilar et al., 2018) also use auxiliary tasks as regularizers but instead of binary classification, the additional tasks performs multi-class classification into B (first word of entity), I (remaining entity words) and O (non-entity).

The auxiliary tasks in MTL can also be used for extra supervision by concatenating their output label distribution to the representation used by NER (Xiao et al., 2019). Unlike prior work, Xiao et al. (2019) do not use the same representation for the target and auxiliary task. Instead, they build a sub-module called similarity-based auxiliary classifier (SAC). SAC takes as input the original input representation and adds token position embeddings (Vaswani et al., 2017), followed by multiple convolution layers. It maintains two randomly initialized vectors representing entity and non-entity classes. These vectors are combined with an attention layer to get the final word representations. The attention weights are calculated with the multiplicative attention function over the word representation and each of these two vectors. The final word representation is concatenated with the biLSTM representation for NER and the attention weights are used as proxy for probabilities of the word being an entity or not. The loss of both tasks is jointly optimized, with less weight given to entity segmentation over NER. Yu et al. (2018) also add extra supervision, but with a two-step approach. They learn two character-level language models for entity and non-entity words and use their output as a binary feature in NER.

Augenstein and Søgaard (2017) use MTL for NER in scientific texts, adding five auxiliary tasks. They add syntactic chunking, hyperlink prediction, multi-word expression identification, frame target annotation, and semantic super-sense tagging; the first three target segmentation. The auxiliary tasks are used one at a time with the target task. Since the datasets for NER and the auxiliary tasks are

Dataset	Genre	Labels
CoNLL	news	PER, LOC, ORG, MISC
Ontonotes	convers-ations	PERSON, ORG, LOC, EVENT, NORP, LANGUAGE, LAW, MONEY, DATE, TIME, WORK OF ART, PERCENT, QUANTITY, CARDINAL, ORDINAL
BTC	twitter	PER, LOC, ORG
TTC	twitter	PER, LOC, ORG

Table 2: Dataset Genre and Entity Labels

different, at each training step, a random task is chosen, followed by a random training instance.

6.2 Semi-Markov CRFs

Semi-Markov CRFs (Sarawagi and Cohen, 2004) are a variant of linear chain CRFs that capture dependencies between adjacent spans of text instead of adjacent words. The Markov assumption still holds across spans but not within the span. The goal is to find the best possible segmentation into spans using scores at span-level. The maximum length of spans is bound to reduce computation cost. Sarawagi and Cohen (2004) use hand-crafted features for span representations but recent work has explored other techniques to represent spans.

Gated recursive semi-markov CRF (Zhuo et al., 2016) creates a pyramid-shaped feature extractor for spans. The bottom-most layer consists of word representations and hence length one spans. Representation of adjacent words are combined to form length two spans for the next layer and so on. The top layer consists of a single span with the full sentence. Hybrid semi-Markov CRF or HSCRF (Ye and Ling, 2018) do not use an explicit span representation. Instead they consider the span-level score as sum of the word-level CRF scores of constituent words. Both the word-level and span-level CRF are jointly optimized. Sato et al. (2017) use a two step process to reduce the search space of the spans. They first generate possible spans from a separate model using a score cutoff and then find the best possible labelling over these spans instead of all spans upto a maximum specified length.

7 Datasets

We evaluate several of the above models on four datasets, to compare their performance. Table 2 shows the entity types in each dataset.

1. **CoNLL** is the English portion of the CoNLL'03 data (Tjong Kim Sang and De Meulder, 2003), extracted from the Reuters 1996 newswire corpus.
2. **ON** is the union of broadcast conversation (bc) and telephone conversation (tc) domains in the English portion of Ontonotes (Hovy et al., 2006).[2] The number of entity types is the largest in this dataset. We merge the closely related categories, *GPE* with *LOC* and *FAC* with *ORG*, to allow us to easily map gazetteer labels to dataset labels.
3. **BTC** or Broad Twitter Corpus (Derczynski et al., 2016) consists of tweets. We use the recommended train, validation and test splits.
4. **TTC** or Temporal Twitter Corpus (Rijhwani and Preotiuc-Pietro, 2020) also consists of tweets. It has multiple training splits from years ranging from 2014 to 2018 and validation and test splits from 2019. We use the 2014 training split as it overlaps less with 2019 and hence is more challenging.

Some dataset statistics are shown in Table 3. CoNLL and OntoNotes are the largest and are roughly equal in size. OntoNotes has more long entities and fewer entities that are sequences of capitalized words. Such characteristics will favor methods that perform better segmentation without relying on standard orthography. BTC and TTC are smaller and have more distinct surface forms. The entities in TTC follow capitalization conventions but BTC has many entities that consist of words that are not capitalized. BTC also exhibits a different distribution of capitalization patterns between the training and the test set. In the training set, roughly half of the entities are sequences of word with capitalized first letter but this number falls to just 28% in the test set. Most entities in all datasets consist of a single words. Only about 2% of entities have length more than three. OntoNotes contains the largest percentage of long entities, 10% of all.

We also present statistics on in/out of vocabulary words in the test data, with respect to the pre-training data, the training data and the gazetteer entries. An entity is considered *seen* in pre-training if the phrase is seen as such in the pre-training corpus or all of the constituent words are seen. For CoNLL and OntoNotes, almost all entities are seen in the pretraining data (>90%), followed by TTC (88%). For BTC, only 65-75% entities are seen in pretraining. Adding ELMo increases the entity coverage by only a small amount in all datasets.

[2]We also experimented with newswire, broadcast news and magazines, which had similar results to CoNLL.

System	CoNLL		ON		BTC		TTC	
	train	test	train	test	train	test	train	test
Entities (spans)	23326	5613	10419	1972	8774	4376	1175	1539
Distinct surface forms (%)	35.49	48.03	37.98	44.62	55.20	59.53	83.83	75.18
Dataset Entities by length in words(%)								
1	62.60	62.94	57.49	47.16	75.88	87.27	68.51	64.72
2	31.51	31.32	22.38	27.74	19.49	10.03	24.94	30.54
3	4.17	4.56	10.62	13.74	2.89	1.97	4.26	3.44
≥ 4	1 .70	1.17	9.51	11.36	1.72	0.73	2.32	1.28
Seen entities (%)								
Pretraining (GloVe)	92.96	92.25	97.13	97.77	75.21	65.40	84.26	87.91
Pretraining (GloVe+ELMo)	95.04	94.53	97.83	98.17	75.92	66.06	84.77	88.56
Training, any type	100.00	62.62	100.00	74.80	100.00	45.70	100.00	22.29
Gazetteer, any type	82.52	82.31	92.73	94.93	65.56	59.32	68.34	79.08
Gaz, not train, any type	0.00	26.55	0.00	21.86	0.00	14.79	0.00	57.76
Training, correct type	100.00	59.43	100.00	69.78	100.00	44.42	100.00	18.91
Gazetteer, correct type	74.34	73.97	67.22	69.78	50.24	32.38	64.09	75.37
Casing								
Title case (%)	82.60	80.76	58.13	56.09	49.50	28.88	67.66	78.17

Table 3: Dataset Statistics. CoNLL and Ontonotes are larger but TTC and BTC have more distinct surface forms. Except Ontonotes, all datasets have very few long entities. Most entities are seen in the pre-training data. Test entities seen in the training data are generally seen with the same type at least. BTC has many lowercase entities.

The gazetteer provides a better coverage than the training data. Training data coverage is especially low in TTC, making the dataset more challenging. According to our definition of seen, entities may be seen in the training data but not necessarily with the expected types. We also check if they are seen in the training data with the same type as in the test data. There is a decrease of only 1-5% when taking the type into consideration. Overwhelming, the type in training is also that in testing.

8 Experiments

Results for models using different word representations, without any gazetteer or segmentation features, are shown in Table 4. We report micro-F1 over all entity types, averaged over three runs. On all datasets, the combination of GloVe, ELMo and character-based representations works best. Given these results, it would have been reasonable to study methods of adding gazetteers only for this representation. However, given that the GloVe along with character-based embeddings is much more commonly used in recent work on NER, we also present results for that.

We now compare one representative approach from each class of methods for adding gazetteer and segmentation features to the biLSTM-CRF architecture. For a fair comparison, we add only the core idea of the model and use the same hyperparameters, noted in the appendix, removing different peripheral features such as part-of-speech tags and word shape, used in papers that introduced the idea.

System	CoNLL	ON	BTC	TTC
G	89.55	77.66	72.04	50.11
G+ch	90.64	77.95	72.75	57.17
E	91.02	79.93	73.33	63.87
E+G+ch	**91.74**	**80.67**	**73.84**	**64.88**

Table 4: NER F1 of models with varying input representation to biLSTM-CRF. G refer to GloVe, E refers to ELMo and ch refers to character-based representation. The highest value in each column is boldfaced.

8.1 Gazetteers

We compare four gazetteer-derived features. In each case, the gazetteer representation vector is passed through a feedforward layer with 32 neurons and ReLU activation and then concatenated to the input word representation to the biLSTM-CRF.

1. **WORD_GAZ** We map gazetteer entity types to dataset types and split gazetteer entries into words using space as a delimiter, to create a vocabulary associated with each entity types, i.e. a list of words that appeared in person names etc. Each word in a sentence is associated with a binary valued vector with length equal to the number of entity types in the dataset. The dimension corresponding to a given type gets value one if the word is in the gazetteer vocabulary for that type and zero otherwise. The vector is all zeros for words that do not appear in the vocabulary for any entity types and can have multiple components with value one, when the word appeared in gazetteer entries of more than one type.

System	CoNLL	ON	BTC	TTC	avg diff	max diff
G+ch	90.64	77.95	72.75	57.17	-	-
WORD_GAZ	90.56	78.21	73.57	58.42	0.56	1.25
GAZ_IOBES	90.44	77.83	73.70	59.70	0.79	2.53
GAZ_FINE	90.52	**79.01**	**73.53**	**58.82**	**0.84**	1.65
PHRASE_GAZ	90.29	77.32	72.73	59.18	0.25	2.01
LRN_GAZ	89.82	78.55	72.91	59.36	0.53	2.19
SEG_REG	90.28	78.46	72.74	56.78	-0.06	0.51
SEG_SUP	90.20	**78.34**	**72.85**	**58.54**	**0.35**	1.37
SAC	90.35	77.33	72.83	57.04	-0.24	0.08
HSCRF	89.91	78.42	71.75	55.34	-0.77	0.47

Table 5: NER F1 on using GloVe+char. *avg-diff* and *max-diff* are the average and maximum increase over the base model across datasets. The most stable system (highest avg-diff) in each category is boldfaced.

System	CoNLL	ON	BTC	TTC	avg diff	max diff
E+G+ch	91.74	80.67	73.84	64.88	-	-
WORD_GAZ	91.74	**80.86**	**74.70**	**66.22**	**0.60**	1.34
GAZ_IOBES	91.76	80.66	72.96	64.60	-0.29	0.02
GAZ_FINE	91.92	80.47	75.03	65.23	0.38	1.19
PHRASE_GAZ	92.08	80.52	74.00	64.19	-0.08	0.34
LRN_GAZ	91.32	81.27	73.98	65.75	0.30	0.87
SEG_REG	91.79	80.99	74.42	63.52	-0.10	0.58
SEG_SUP	91.81	80.66	74.97	65.51	0.46	1.13
SAC	91.87	80.20	75.27	62.57	-0.30	1.43
HSCRF	91.66	80.62	**74.64**	**66.92**	**0.68**	2.04

Table 6: NER F1 using ELMo+GloVe+char. *avg-diff* and *max-diff* are the average and maximum increase over the base model across datasets. The most stable system (highest avg-diff) in each category is boldfaced.

2. **GAZ_IOBES** is similar to WORD_GAZ but instead of a single bit, there are four values for each label denoting a matching to an entity exactly (S), to the first word (B), to the last word (E) or other words in between (I).

3. **GAZ_FINE** is also similar to WORD_GAZ but instead of selecting types from the gazetteer to match the label types for a given dataset, we use all 30 gazetteer types as is. The length of the gazetteer vector equals the numbers of the original gazetteer types without any mapping.

4. **PHRASE_GAZ** Here, we perform phrase level matching between the sentence and the gazetteer. All subsequences (continuous sequence of words) in a sentence are matched to each entry in the gazetteers, retaining the gazetteer type only for the longest match. For example, only 'New York University' is matched, not the nested 'New York' entity in it. Each word in the longest match subsequence gets the bit corresponding to the matched gazetteer category switched on in the gazetteer vector representation.

5. **LRN_GAZ** Gazetteer embeddings are learned with a separate model and concatenated to the input representation for the NER model, as in Magnolini et al. (2019). Random n-grams, excluding named entities, from CoNLL'03 training data are used as negative examples for training the gazetteer network, as opposed to negative examples generated from gazetteer entries used in Magnolini et al. (2019).

Both word-level matching and learned gazetteers make it possible for the model to learn phrases beyond the exact entries in the gazetteers. For example, if the organization gazetteer contains 'GAZ High School' but the dataset contains 'TST High School', phrase-level matching would not match any of the words in 'TST High School' to a gazetteer. A word-level matching would at least mark 'High' and 'School' as organizations. Similarly, learned gazetteer embeddings may learn a similar representation for both and make it possible to recognize 'TST High School' correctly.

# gaz types	CoNLL	ON	BTC	TTC
0	61.9	59.2	74.5	62.8
1	15.7	13.9	10.1	14.3
1+	21.7	26.9	6.3	9.0
all	0.7	0.0	9.1	13.8

Table 7: Percentage of words in the vocabulary with the number of gazetteer types in which they appear.

8.2 Segmentation

We experiment with the different multi-task learning methods of incorporating segmentation and one of the semi-Markov CRF models.

1. **SEG_REG** (Aguilar et al., 2017) is the MTL approach with two auxiliary tasks: binary classification to predict if the word is a part of entity and multi-class classification to predict the type of the word without CRF.

2. **SEG_SUP** is the same as SEG_REG, but the label distribution scores from the binary classification auxiliary task are concatenated to the final representation used for NER.

3. **SAC** is the similarity-based auxiliary classifier in Xiao et al. (2019) as described above.

4. **HSCRF** is the hybrid semi-Markov CRF in Ye and Ling (2018) as described above

System	CoNLL	ON	BTC	TTC	avg diff	max diff
G+ch	94.84	83.24	85.34	67.74	-	-
WORD_GAZ	94.86	83.02	86.60	69.20	0.63	1.46
GAZ_IOBES	94.90	83.03	86.50	70.15	0.86	2.41
GAZ_FINE	**94.85**	**83.67**	**86.07**	**70.17**	**0.90**	2.43
PHRASE_GAZ	94.74	82.26	86.00	69.64	0.37	1.90
LRN_GAZ	94.44	83.65	85.70	68.97	0.40	1.23
SEG_REG	94.80	83.66	85.75	66.90	-0.01	0.42
SEG_SUP	94.71	83.16	**86.20**	**69.98**	**0.72**	2.24
SAC	94.79	82.74	86.93	69.25	-0.42	-0.19
HSCRF	94.55	82.11	85.15	67.03	0.48	1.59

Table 8: Entity segmentation F1 using GloVe+char. *avg-diff* and *max-diff* are the average and maximum increase over the base model across datasets. The most stable system (highest avg-diff) in each category is boldfaced.

System	CoNLL	ON	BTC	TTC	avg diff	max diff
E+G+ch	95.48	85.01	85.4	73.59	-	-
WORD_GAZ	95.32	**85.17**	**86.67**	**74.29**	**0.49**	1.27
GAZ_IOBES	95.54	84.77	85.89	73.64	0.09	0.49
GAZ_FINE	95.51	84.61	86.55	74.10	0.32	1.15
PHRASE_GAZ	95.54	84.64	85.84	73.17	-0.07	0.44
LRN_GAZ	95.08	85.18	86.07	74.52	0.34	0.93
SEG_REG	95.60	85.08	86.46	72.88	0.13	1.06
SEG_SUP	95.51	84.76	86.38	74.57	0.43	0.98
SAC	95.37	84.38	86.93	71.70	-0.28	1.53
HSCRF	95.37	84.73	**86.38**	**76.63**	**0.91**	3.04

Table 9: Entity Segmentation F1 using ELMo+GloVe+char. *avg-diff* and *max-diff* are the average and maximum increase over the base model across datasets. The most stable system (highest avg-diff) in each category is boldfaced.

9 Results

We test the gazetteer-enhanced and segmentation approaches using both GloVe+char and ELMo+GloVe+char as the input representations. Results are shown in Tables 5 and 6 respectively.

9.1 Consistency Across Datasets

We report the average and maximum improvement in F1-score over the base model across datasets. A high average improvement means that the model is consistently better across datasets spanning newswire, conversations and Twitter posts. A high maximum improvement with a low or negative average improvements means that the model can do well on some dataset but fails to perform well when tested on multiple datasets of varied genres.

The word indicator gazetteer features are the best and most consistent on average. With ELMo+GloVe, they also show the maximum improvement. WORD_GAZ combines gazetteer labels to dataset labels whereas the GAZ_FINE does not do any dataset specific mapping. Prior work has used the former but this experiment shows that we do not need to do such dataset specific modifications of gazetteer labels. We can use fine-grained labels and obtain similar gains. PHRASE_GAZ and LRN_GAZ improve performance with GloVe+char, with especially high performance on TTC, but they are not as good with ELMo+GloVe+char. Improvement with both representation by incorporating gazetteers is higher on BTC and TTC than CoNLL and OntoNotes. This is likely because the gazetteer features are more ambiguous in CoNLL and OntoNotes (Table 7), with most words appearing in two or more possible categories.

Segmentation methods can show vast improvement on specific datasets but they are unstable across datasets. The average improvement is *negative* in many cases for both representations. SEG_SUP is the only segmentation approach that is consistently beneficial across datasets for both representations. HSCRF performs well too, but only when ELMo is used as well. In comparison, including gazetteers is consistently better.

9.2 Gazetteers for Segmentation

In this section, we explore if gazetteers can be used as an alternative for segmentation, owing to their stable performance across datasets. We ask the following question:Are gazetteers recognizing new entities not previously marked as entities or are they improving the typing of spans already recognized as entities by adding entity type information? To answer this question, we report the entity segmentation F1 in Tables 8 and 9. Entity segmentation is the task of finding the correct entity span, regardless of the type. Since pre-training data covers most entities (Table 3), one would expect gazetteers to improve typing of entities through a more explicit type signal. However, segmentation results are consistent with those of NER. The models that improved performance of NER also perform well on segmentation and often by similar margins as NER. In fact, following similar trends as NER, gazetteer-enhanced models are more consistent than segmentation methods at entity segmentation as well. Since gazetteer-enhanced models induce segmentation and improve performance despite a near perfect coverage provided by the pre-training data, we expect them to improve performance of even the latest transformer-based models

System	CoNLL				Ontonotes				BTC				TTC			
	1	2	3	≥ 4	1	2	3	≥ 4	1	2	3	≥ 4	1	2	3	≥ 4
E+G+ch	92.1	92.0	**85.8**	**85.4**	**87.0**	78.7	64.4	74.3	74.2	**75.5**	57.6	41.3	59.9	77.6	54.0	38.8
WORD_GAZ	92.1	92.0	85.5	85.0	**87.0**	78.4	**65.6**	**75.6**	75.2	75.2	**58.0**	**44.6**	61.5	78.3	**54.5**	35.8
HSCRF	**92.2**	**92.2**	83.8	78.9	86.8	**79.3**	63.5	73.7	**75.3**	75.4	53.2	36.9	**62.1**	**79.4**	50.7	**47.2**

Table 10: NER F1 by entity length (words) using ELMo+GloVe+char on the most stable gazetteer and segmentation method. The highest value in each column is bolfaced. Generally, gazetteers perform better on long entities and segmentation methods perform better on short entities.

System	CoNLL	ON	BTC	TTC	avg diff	max diff
G+ch	90.64	77.95	72.75	57.17	-	-
WORD_GAZ	90.56	78.21	73.57	58.42	0.56	1.25
- gaz types	90.64	78.69	74.13	59.16	1.03	1.99
E+G+ch	91.74	80.67	73.84	64.88	-	-
WORD_GAZ	91.74	80.86	74.70	66.22	0.60	1.34
- gaz types	91.58	80.31	74.75	65.16	0.17	0.91

Table 11: NER F1. *avg-diff* and *max-diff* are the average and maximum increase over the base model across datasets. Removing gazetteer typ information improves performance for GloVe+char.

(Devlin et al., 2019) pre-trained on large corpora.

Next, we modify WORD_GAZ to mark the presence in any gazetteer without taking the type into consideration (Table 11). Surprisingly, removing type information results in better performance with GloVe, indicating the most benefit came from segmentation and not typing. With ELMo included, however, we do not observe the same trend. While performance is better than the baseline system, it is not better than WORD_GAZ that includes types. Though we cannot point conclusively to the reason behind such results with ELMo, we suspect it is due to the ambiguity within the pre-training data. The model may not be using the gazetteer representation if there is a strong signal from the pretrained representation that the word is not an entity. This can only be verified if all representations are trained on the same pre-training corpus.

9.3 Gazetteers for Long Entities

To further verify the effectiveness of gazetteers for segmentation, we break down performance by *entity length* in words, reporting F1 in Table 10 for the top performing gazetteer-enhanced model and segmentation model with ELMo+GloVe+char. Recall that the majority of entities in all corpora are of length one and that entities consisting of more than three words are the rare, about 2% in three of the datasets, expect in OntoNotes, where they are 10%. The highest performance is on entities of length two, followed by length one. Longer entities are recognized much more more poorly.

Typically segmentation methods have been used to improve performance on long entities. But our experiments reveal that gazetteers are better at it. With the exception of TTC, WORD_GAZ is better on long entities and HSCRF on shorter ones. This is likely due to the presence of many long entities in gazetteers (§4, Table 1).

10 Conclusion

We provided a comprehensive overview of methods for incorporating gazetteers and inducing segmentation in NER. We chose these two areas because even though they have been explored separately in prior work, we find that they are interrelated and achieve similar goals. We implemented representative models from each category for a fair comparison. We found that while segmentation methods can achieve impressive improvements on specific datasets, gazetteer-enhanced models are more stable across datasets. Moreover, the simpler methods of gazetteer enhancement (binary valued discrete feature vector with word-level gazetteer matching) and segmentation (multi-task learning with a extra supervision from and auxiliary binary classification for segmentation) performed better within their respective categories.

Furthermore, contrary to expectation, we found that gazetteer-enhanced models improve entity segmentation, not just entity typing. In fact, one need not perform a gazetteer to dataset label mapping for incorporating gazetteers; using the original gazetteer types works just as well. Even more surprisingly, gazetteer types are even unnecessary depending on the input representation. With GloVe, performance improves by removing gazetteer types altogether. This is likely a consequence of gazetteers inducing segmentation. Lastly, we showed that gazetteers are better at finding long entities, another consequence of inducing segmentation. They are an effective alternative to segmentation techniques developed to identify long entities, which we found are unstable across datasets.

References

Oshin Agarwal, Yinfei Yang, Byron C. Wallace, and A. Nenkova. 2020. Entity-switched datasets: An approach to auditing the in-domain robustness of named entity recognition models. *ArXiv*, abs/2004.04123.

Oshin Agarwal, Yinfei Yang, Byron C. Wallace, and Ani Nenkova. 2021. Interpretability Analysis for Named Entity Recognition to Understand System Predictions and How They Can Improve. *Computational Linguistics*, 47(1):117–140.

Gustavo Aguilar, Adrian Pastor López-Monroy, Fabio González, and Thamar Solorio. 2018. Modeling noisiness to recognize named entities using multitask neural networks on social media. 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 1401–1412, New Orleans, Louisiana. Association for Computational Linguistics.

Gustavo Aguilar, Suraj Maharjan, Adrian Pastor López-Monroy, and Thamar Solorio. 2017. A multi-task approach for named entity recognition in social media data. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 148–153, Copenhagen, Denmark. Association for Computational Linguistics.

Isabelle Augenstein, Mrinal Das, Sebastian Riedel, Lakshmi Vikraman, and Andrew McCallum. 2017a. SemEval 2017 task 10: ScienceIE - extracting keyphrases and relations from scientific publications. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 546–555, Vancouver, Canada. Association for Computational Linguistics.

Isabelle Augenstein, Leon Derczynski, and Kalina Bontcheva. 2017b. Generalisation in named entity recognition: A quantitative analysis. *ArXiv*, abs/1701.02877.

Isabelle Augenstein and Anders Søgaard. 2017. Multi-task learning of keyphrase boundary classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 341–346, Vancouver, Canada. Association for Computational Linguistics.

Oliver Bender, Franz Josef Och, and Hermann Ney. 2003. Maximum entropy models for named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 148–151.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. 2014. One billion word benchmark for measuring progress in statistical language modeling. In *Fifteenth Annual Conference of the International Speech Communication Association*.

Colin Cherry and Hongyu Guo. 2015. The unreasonable effectiveness of word representations for Twitter named entity recognition. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 735–745, Denver, Colorado. Association for Computational Linguistics.

Jason P.C. Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics*, 4:357–370.

Michael Collins and Yoram Singer. 1999. Unsupervised models for named entity classification. In *1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(ARTICLE):2493–2537.

Xiang Dai, Sarvnaz Karimi, Ben Hachey, and Cecile Paris. 2019. Using similarity measures to select pretraining data for NER. 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 1460–1470, Minneapolis, Minnesota. Association for Computational Linguistics.

Leon Derczynski, Kalina Bontcheva, and Ian Roberts. 2016. Broad Twitter corpus: A diverse named entity recognition resource. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1169–1179, Osaka, Japan. The COLING 2016 Organizing Committee.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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, Minneapolis, Minnesota. Association for Computational Linguistics.

Rezarta Islamaj Doğan, Robert Leaman, and Zhiyong Lu. 2014. Ncbi disease corpus: a resource for disease name recognition and concept normalization. *Journal of biomedical informatics*, 47:1–10.

Doug Downey, Matthew Broadhead, and Oren Etzioni. 2007. Locating complex named entities in web text. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, page 2733–2739, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Jacob Eisenstein. 2013. What to do about bad language on the internet. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 359–369, Atlanta, Georgia. Association for Computational Linguistics.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 363–370, Ann Arbor, Michigan. Association for Computational Linguistics.

JinLan Fu, Peng fei Liu, Qi Zhang, and Xuanjing Huang. 2020a. Rethinking generalization of neural models: A named entity recognition case study. *ArXiv*, abs/2001.03844.

Jinlan Fu, Pengfei Liu, and Graham Neubig. 2020b. Interpretable multi-dataset evaluation for named entity recognition. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6058–6069, Online. Association for Computational Linguistics.

Octavian-Eugen Ganea and Thomas Hofmann. 2017. Deep joint entity disambiguation with local neural attention. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2619–2629, Copenhagen, Denmark. Association for Computational Linguistics.

Abbas Ghaddar and Phillippe Langlais. 2018. Robust lexical features for improved neural network named-entity recognition. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1896–1907, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Ralph Grishman and Beth Sundheim. 1996. Message Understanding Conference- 6: A brief history. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*, pages 466–471.

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

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. OntoNotes: The 90% solution. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 57–60, New York City, USA. Association for Computational Linguistics.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

J-D Kim, Tomoko Ohta, Yuka Tateisi, and Jun'ichi Tsujii. 2003. Genia corpus—a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(suppl_1):i180–i182.

Martin Krallinger, Obdulia Rabal, Florian Leitner, Miguel Vazquez, David Salgado, Zhiyong Lu, Robert Leaman, Yanan Lu, Donghong Ji, Daniel M Lowe, et al. 2015. The chemdner corpus of chemicals and drugs and its annotation principles. *Journal of cheminformatics*, 7(1):1–17.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California. Association for Computational Linguistics.

Tianyu Liu, Jin-Ge Yao, and Chin-Yew Lin. 2019a. Towards improving neural named entity recognition with gazetteers. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5301–5307, Florence, Italy. Association for Computational Linguistics.

Yijin Liu, Fandong Meng, Jinchao Zhang, Jinan Xu, Yufeng Chen, and Jie Zhou. 2019b. GCDT: A global context enhanced deep transition architecture for sequence labeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2431–2441, Florence, Italy. Association for Computational Linguistics.

Teng Long, Ryan Lowe, Jackie Chi Kit Cheung, and Doina Precup. 2016. Leveraging lexical resources for learning entity embeddings in multi-relational data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 112–117, Berlin, Germany. Association for Computational Linguistics.

Xuezhe Ma and Eduard Hovy. 2016. 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, Berlin, Germany. Association for Computational Linguistics.

Simone Magnolini, Valerio Piccioni, Vevake Balaraman, Marco Guerini, and Bernardo Magnini. 2019. How to use gazetteers for entity recognition with neural models. In *Proceedings of the 5th Workshop on Semantic Deep Learning (SemDeep-5)*, pages 40–49, Macau, China. Association for Computational Linguistics.

Xue Mengge, Bowen Yu, Zhenyu Zhang, Tingwen Liu, Yue Zhang, and Bin Wang. 2020. Coarse-to-Fine Pre-training for Named Entity Recognition. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6345–6354, Online. Association for Computational Linguistics.

Andrei Mikheev, Marc Moens, and Claire Grover. 1999. Named entity recognition without gazetteers. In *Ninth Conference of the European Chapter of the Association for Computational Linguistics*, pages 1–8, Bergen, Norway. Association for Computational Linguistics.

Scott Miller, Jethran Guinness, and Alex Zamanian. 2004. Name tagging with word clusters and discriminative training. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 337–342, Boston, Massachusetts, USA. Association for Computational Linguistics.

Einat Minkov, Richard C. Wang, and William W. Cohen. 2005. Extracting personal names from email: Applying named entity recognition to informal text. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 443–450, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Naoaki Okazaki. 2007. Crfsuite: a fast implementation of conditional random fields (crfs).

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Duangmanee Putthividhya and Junling Hu. 2011. Bootstrapped named entity recognition for product attribute extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1557–1567, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, pages 147–155, Boulder, Colorado. Association for Computational Linguistics.

Shruti Rijhwani and Daniel Preotiuc-Pietro. 2020. Temporally-informed analysis of named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7605–7617, Online. Association for Computational Linguistics.

Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. 2011. Named entity recognition in tweets: An experimental study. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1524–1534, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Sunita Sarawagi and William W Cohen. 2004. Semimarkov conditional random fields for information extraction. *Advances in neural information processing systems*, 17:1185–1192.

Motoki Sato, Hiroyuki Shindo, Ikuya Yamada, and Yuji Matsumoto. 2017. Segment-level neural conditional random fields for named entity recognition. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 97–102, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Dominic Seyler, Tatiana Dembelova, Luciano Del Corro, Johannes Hoffart, and Gerhard Weikum. 2018. A study of the importance of external knowledge in the named entity recognition task. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 241–246, Melbourne, Australia. Association for Computational Linguistics.

Chan Hee Song, Dawn Lawrie, Tim Finin, and James Mayfield. 2020. Improving neural named entity recognition with gazetteers. *arXiv preprint arXiv:2003.03072*.

Karl Stratos. 2017. Entity identification as multitasking. In *Proceedings of the 2nd Workshop on Structured Prediction for Natural Language Processing*, pages 7–11, Copenhagen, Denmark. Association for Computational Linguistics.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30:5998–6008.

Shiyuan Xiao, Yuanxin Ouyang, Wenge Rong, Jianxin Yang, and Zhang Xiong. 2019. Similarity based auxiliary classifier for named entity recognition. 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 1140–1149, Hong Kong, China. Association for Computational Linguistics.

Eunsuk Yang, Young-Bum Kim, Ruhi Sarikaya, and Yu-Seop Kim. 2016. Drop-out conditional random fields for Twitter with huge mined gazetteer. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 282–288, San Diego, California. Association for Computational Linguistics.

Zhixiu Ye and Zhen-Hua Ling. 2018. Hybrid semi-Markov CRF for neural sequence labeling. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 235–240, Melbourne, Australia. Association for Computational Linguistics.

Xiaodong Yu, Stephen Mayhew, Mark Sammons, and Dan Roth. 2018. On the strength of character language models for multilingual named entity recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3073–3077, Brussels, Belgium. Association for Computational Linguistics.

Jingwei Zhuo, Yong Cao, Jun Zhu, Bo Zhang, and Zaiqing Nie. 2016. Segment-level sequence modeling using gated recursive semi-Markov conditional random fields. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1413–1423, Berlin, Germany. Association for Computational Linguistics.

A Dataset Label to Gazetteer Mapping

CoNLL

- *PER*: People
- *LOC*: Locations, Locations.Generic, Parks
- *ORG*: Organization, PoliticalParties, Corporations, Government
- *MISC*: EthnicGroups, Languages, Vehicles, Nationalities

Ontonotes

- *PERSON*: People
- *LOC*: Locations, Locations.Generic, Parks
- *ORG*: Organization, PoliticalParties, Corporations, Government
- *DATE*: Temporal
- *TIME*: Temporal
- *CARDINAL*: NumberCardinal
- *ORDINAL*: NumberOrdinal
- *EVENT*: CompetitionsBattlesEvents
- *LANGUAGE*: Languages
- *MONEY*: Currency
- *NORP*: Nationalities, EthnicGroups
- *QUANTITY*: Measurements
- *WORK_OF_ART*: TV.Programs, ArtWork, Films

BTC and TTC

- *PER*: People
- *LOC*: Locations, Locations.Generic, Parks
- *ORG*: Organization, PoliticalParties, Corporations, Government

B Implementation

We use the implementation in https://github.com/guillaumegenthial/tf_ner for most models.

SAC uses https://github.com/XiaoShiyuan/NCRF-SAC but is based off the same codebase.

HSCRF uses https://github.com/ZhixiuYe/HSCRF-pytorch but ELMo was added additionally by us.

C Hyperparameters

Hyperparameters used as same as the ones in https://github.com/guillaumegenthial/tf_ner/models/chars_conv_lstm_crf, except for number of epochs and the minimum number of steps before early stopping which depend on dataset size. Numbers of epochs used were 25, 25, 50 and 50 for CoNLL, Ontonotes, BTC and TTC respectively. Minimum steps were 8000, 15000, 2500 and 2500 for CoNLL, Ontonotes, BTC and TTC respectively.