How to Use Gazetteers for Entity Recognition with Neural Models

Simone Magnolini[1], **Valerio Piccioni**[1,2], **Vevake Balaraman**[1,2], **Marco Guerini**[1], **Bernardo Magnini**[1]
[1] Fondazione Bruno Kessler, Via Sommarive 18, Povo, Trento — Italy
[2] University of Trento, Italy.
{magnolini, balaraman, guerini, magnini}@fbk.eu
valerio.piccioni@studenti.unitn.it

Abstract

Although the use of end-to-end neural architectures has been proven to be effective on several sequence labeling tasks, the use of gazetteers in these architectures is still rather unexplored. We investigate several options, aiming at exploiting gazetteers to extract relevant features, and then at integrating these features in a neural model for entity recognition. We provide experimental evidences on two datasets (named entities and nominal entities) and two languages (English and Italian), showing that extracting features from a rich model of the gazetteer and then concatenating such features with the input embeddings of a neural model is the best strategy in all our experimental settings, significantly outperforming more conventional approaches.

1 Introduction

In the recent years a number of neural architectures have been successfully applied to several sequence labelling tasks, including, among others, part-of-speech tagging (Choi, 2016), named entity recognition (Ma and Hovy, 2016), and semantic role labeling (He et al., 2017). It has been shown that these architectures can achieve state-of-art performance with an end-to-end configuration, i.e. without recurring either to linguistic features or to external knowledge sources (e.g. gazetteers). However, experiments have been often conducted over datasets with large amount of training data and in a rather limited spectrum of experimental conditions. Overall, we think that there has not been much discussion about the use of gazetteers together with neural models, and that a deeper investigation is necessary.

In this paper we focus on the role of gazetteers for entity recognition. The following are our two main research questions: (i) As neural networks architectures are highly modular, which is the best way to integrate information from gazetteers? (ii) What is the impact of the size of both training data and gazetteers over the performance of a neural model for entity recognition?

As mentioned, we focus on entity recognition and refer to the Automatic Content Extraction program - ACE (Doddington et al., 2004). In this context, entity recognition has been approached as a sequence labeling task. Given an utterance $U = \{t_1, t_2, ..., t_n\}$ and a set of entity categories $C = \{c_1, c_2, ..., c_m\}$, the task is to label the tokens in U that refer to entities belonging to the categories in C. As an example, using the IOB format (Inside-Outside-Beginning, (Ramshaw and Marcus, 1995)), the sentence “I would like to order a salami pizza and two mozzarella cheese sandwiches” could be labeled as shown in Table 1. ACE distinguishes two main entity classes: *named entities* and *nominal entities*, and we consider both of them for our experiments.

The first entity class, named entities, roughly corresponds to proper names, and named entities recognition (NER) tools for frequent categories (i.e. persons as “Barack Obama”, locations as “New York”, and organizations as “IBM”) have been developed for many languages. Several datasets are available for training purposes (e.g. the Conll-2003 datasets (Tjong Kim Sang and De Meulder, 2003)). It has been a common practice of NER systems to make use of gazetteers (i.e. lists of entity names), considering the presence of a token in certain gazetteer as an additional feature for the classifier (see, for instance, the use of the Stanford NER $useGazettes$ parameter for the CRF classifier (Finkel et al., 2005)).

Nominal entities, on the other hand, are noun phrase expressions describing an entity. Differently from named entities, nominal entities are typically compositional, as they do allow morphological and syntactic variations (e.g. for food

I	would	like	to	order	a	salami	pizza	and	two	mozzarella	cheese	sandwiches
O	O	O	O	O	O	B-FOOD	I-FOOD	O	O	B-FOOD	I-FOOD	I-FOOD

Table 1: IOB annotation of food entities inside user request.

names, *spanish baked salmon*, *roasted salmon* and *hot smoked salmon*), which makes it possible to combine tokens of one entity name with tokens of another entity name to generate new names (e.g. for food names, *salmon tacos* is a potential food name given the existence of *salmon* and *tacos*). In the framework of the ACE program there have been several attempts to develop supervised systems for nominal entities (Biggio et al., 2010); these systems, however, had to face the problem of the scarcity of annotated data, and, for this reason, were developed for few entity types.

In this paper we make use of an end-to-end state-of-art entity recognition system (described in Section 2), and investigate the combination with gazetteers under several integration methods, which are described in Section 3 and 4. Datasets for our experiments are described in Section 5, while results are presented and discussed in Section 6.

2 Core Entity Recognition System

In order to investigate the use of gazetteers in combination with neural models we first need an entity recognition system. For our experiments we have adopted the neural system proposed in (Ma and Hovy, 2016), which achieved state-of-art performance for named entity recognition for English on the ConLL-2003 dataset (see section 5). Specifically, we use the most recent implementation of the system in Pytorch distributed by the authors[1], and called *NeuroNLP2*. The system is composed of three layers (Figure 1): (i) a CNN that allows to extract information from the input text without any pre-processing; (ii) a bidirectional LSTM layer that presents each sequence forwards and backwards to two separate LSTM; (iii) a CRF layer that decodes the best label sequence.

For each token in the input sequence, first a character-level representation is computed by a CNN with character embeddings as inputs. Then the character-level representation vector is concatenated with the word embedding vector to feed the BLSTM network. The CNN for Character-level Representation is an effective approach to extract morphological information (like the prefix or suffix of a word) from characters of words and encode it into neural representations. In NeuroNLP2 the CNN is similar to the one proposed in (Chiu and Nichols, 2015), except that it uses only character embeddings as inputs, without character type.

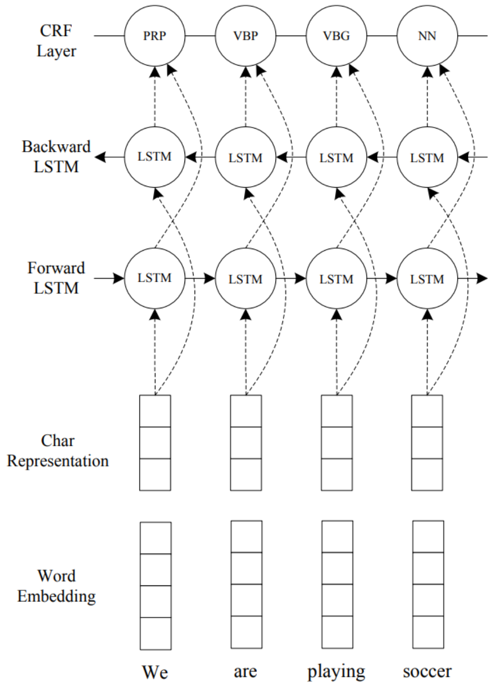

Figure 1: The main NeuroNLP2 structure. Dashed arrows indicate dropout layers applied on both the input and output vectors of BLSTM.

At the second layer each input sequence is presented both forwards and backwards to a bidirectional LSTM, whose output allows to capture past and future information. LSTMs (Hochreiter and Schmidhuber, 1997) are variants of recurrent neural networks (RNNs) designed to cope with gradient vanishing problems. The LSTMs hidden state takes information only from the past, knowing nothing about the future. However, for many tasks it is beneficial to have access to both past (left) and future (right) contexts. A possible solution, whose effectiveness has been proven by previous work (Dyer et al., 2015), is provided by bi-directional LSTMs (BLSTM). (Ma and Hovy, 2016) apply a dropout layer on both the input and output vectors

[1] https://github.com/XuezheMax/NeuroNLP2

of the BLSTM.

Finally, the third layer implemented by NeuroNLP2 is a Conditional Random Fields (CRF) based decoder, which considers dependencies between entity labels in their context and then jointly decodes the best chain of labels for a given input sentence. For example, in NER with standard IOB annotation, an I-token can not follow an O, a constraint which is captured by the CFR layer. Conditional Random Fields (Lafferty et al., 2001) offer several advantages over hidden Markov models and stochastic grammars for such tasks, including the ability to relax strong independence assumptions made in those models. For a sequence CRF model (only interactions between two successive labels are considered), training and decoding can be solved efficiently by adopting the Viterbi algorithm.

We use exactly the same network parameters described in (Ma and Hovy, 2016) and provided as default by the available implementation. As input embeddings we use Stanford's publicly available GloVe 100-dimensional embeddings trained on 6 billion words from Wikipedia and web texts for English (in the same way as (Ma and Hovy, 2016)); for Italian we use Stanford's GloVe 50-dimensional embeddings trained on a Wikipedia's dump [2] with the default setup. For the out of vocabulary words we use a unique randomly generated vector for every word.

3 Gazetteers as Features

In this section we present three methods that allow us to exploit information contained in gazetteers and to represent such information as features to be used by the neural entity recognition system. While the first two methods, single token presence and multi-token presence, have been often used, the third method, i.e. gazetteer neural model, is based on the assumption that a more complex model can better exploit the properties of nominal entities.

3.1 Single Token Presence

A simple and straight-forward approach to use a gazetteer is to consider the presence of a single token in the gazetteer as a feature. To do that, we extract the vocabulary of the gazetteer and provide a boolean value to every token in the sentence, which indicates the presence or absence of the token in the vocabulary. If the number of gazetteers in the domain is n, corresponding to the number of entity classes, a single token takes a vector of n-dimensions: if the vocabularies of the different gazetteers do not overlap this is a one-hot vector, otherwise we can have multiple positive dimensions.

3.2 Multi-token Presence

The second approach uses the same feature space as the single token presence method, but instead of checking for the presence of a single token in the gazetteer, it looks for the longest entity name in the gazetteer contained in the sentence. Let us consider the example in Table 1, *I would like to order a salami pizza and two mozzarella cheese sandwiches*, and assume a gazetteer for the class FOOD composed of two entries: *mozzarella pizza* and *salami sandwiches*. With the multi-token approach none of the tokens would have the gazetteer feature equal to true, while with the single token approach both *salami*, *pizza*, *mozzarella* and *sandwiches* would have the presence set to true. The multi-token technique enables a more consistent usage of gazetteers, especially in case of noisy entity names, although a possible drawback could be a lack of generalization.

3.3 Gazetteer Neural Model: NN$_g$

The third method to extract features from a gazetteer follows the intuition that the presence-absence approaches presented in Sections 3.1 and 3.2 might not be adequate for nominal entities, which show higher linguistic complexity than named entities. The idea is to build a neural classifier trained solely on gazetteers, that classifies a subsequence of tokens on the input sentence as belonging to a certain entity class with a certain confidence. Then we use the output of such classifier as a feature to be integrated within the NeuroNLP2 system.

The neural architecture of the entity classifier, the features it uses, and the methodology to generate synthetic negative examples are briefly presented in the following.

Architecture of the NN$_g$ Classifier. We used the neural gazetteer-based approach (called NN$_g$) proposed by (Guerini et al., 2018). The NN$_g$ classifier is implemented using a multilayer bidirectional LSTM that classifies an input sequence of

[2] 20/04/2018

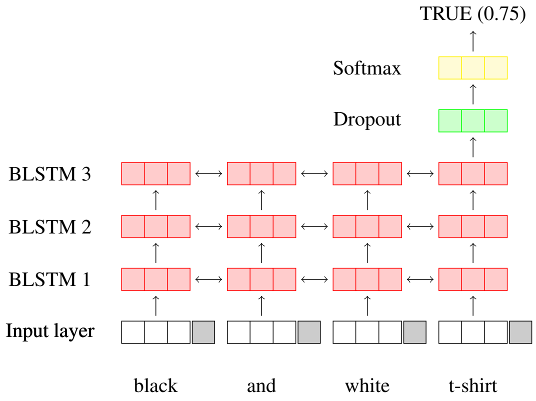

Figure 2: Structure of the neural gazetteer entity recognition (NN_g). The input layer concatenates the features in a single vector.

tokens either as an entity of a certain gazetteer or a non-entity, with a degree of confidence, i.e. the system classifies the sequences in number-of-gazetteers plus one (non-entity) different classes. The NN_g classifier is based on the system proposed in (Lample et al., 2016). The core is still a bidirectional LSTM, but it has a 3-layer BLSTM with 128 units per layer and with a single dropout layer (with a dropout probability of 0.5) between the third BLSTM and the output layer (a softmax layer). The topology of the network is depicted in Figure 2.

NN_g Classifier Features. The NN_g classifier combines several features: word embeddings, char-based embedding, and nine handcrafted features. Word embeddings are similar to those used for NeuroNLP2. For English we used Stanfords publicly available 50-dimensions embeddings, while for Italian we use 50-dimensional embeddings trained on a Wikipedia's dump with the default setup. The char-based embeddings, with a dimension of 30, are based on (Lample et al., 2016), and are trained on the entries in the gazetteers. The expected role of the char-based embeddings is to deal with out of vocabulary terms and possible word misspellings.

The handcrafted features are meant to explicitly represent the structure of a certain entity name, as it can be induced from the gazetteer in which the entity appears. NN_g considers nine features for each token in an entity name: (i) the actual position of the token within an entity name; (ii) the length of the entity name under inspection; (iii) the frequency of the token in the gazetteer; (iv) the average length of the entity names containing a certain token; (v) the average position of the token in the entity names it appears in; (vi) the bigram probability with reference to the previous token in the entity name; (vii) whether the token can be an entity or not; (viii) the ratio of the times the token is the first token in an entity name; (ix) the ratio of the times the token is the last token in an entity name.

Generating Synthetic Training Data. The NN_g classifier for a certain entity class is trained with both positive examples, i.e. entity names present in a gazetteer of the entity class, and negative examples, which are obtained by synthetic generation from the positive examples. In the following we explain how negative examples are generated.

For each entity name i in a gazetteer G (i.e. the positive example), negative counterparts are subsequences of i, or i with additional tokens at the beginning or end of it (or both), e.g. *t1* + i + *t2*. Where *t1* is the ending token of a random entity in G and *t2* is the starting token of a random entity in G. Between these tokens and i there can be separators, as a white space, a comma or the *and* conjunction, so to mimic how multiple entities are usually expressed in sentences. Alternatively, *t1* and *t2* can be tokens randomly extracted from a generic corpus, so to mimic cases when the entity is expressed in isolation.

For example, if the initial positive example is *Community of Madrid*, the possible negative subsequences that are generated are: | *Community* | *of* | *Community of* | *of Madrid* |. The subsequence | *Madrid* | is not considered because it is already included in the gazetteer as positive example. Adding tokens, using the pattern *t1* + i + *t2*, we obtain other potential negative examples: | *contemporary Community of Madrid* | *Community of Madrid and Murcia* | *contemporary Community of Madrid and Murcia* |, and so on. According to this procedure, we generate more negative examples than positive. In order to avoid a too unbalanced datatset, we randomly selected two negative examples for each positive example: a sub-sequence and an example surrounded by other words, resulting in a $1:2$ proportion.

4 Integrating Gazetteer Features

In this section we present the two methods used in our experiments to integrate the features extracted from gazetteers (see Section 3) into NeuroNLP2, the neural entity recognition system. Each of the integration methods adds one boolean feature for each gazetteer.

4.1 Integration 1: Gazetteer Features as Embedding Dimensions

Once we have extracted gazetteer features for each token of the input sentence, the first approach that we consider is to feed such features directly into the neural network. In this method the gazetteer information, represented by a n-dimensions vector, is simply concatenated with the embedding of each token of the input sentence. By default, the NeuroNLP2 system (see Section 2) uses both character and word embeddings, which are concatenated and passed on to the BLSTM layer to learn from them. In this approach the gazetteer feature is concatenated with the character and word embedding, and then it is passed to the BLSTM. The embedding representation for a given token x is as follows:

$$Embedding_x = [x_{word}; x_{char}; x_{gaz}]$$

While the integration as embedding dimensions for the single token and the multi-token features is straightforward, in order to combine NN$_g$ with NeuroNLP2 we need to substitute part of the NN$_g$'s network after training. In fact, we need a NN$_g$'s output for every token, while NN$_g$ classifies a sequence of tokens. To do that we remove the softmax layer of NN$_g$ and we feed the output vectors of the third BLSTM to a fully connected layer of 32 nodes followed by a rectified linear unit (ReLU). With this modification we add to NeuroNLP2 32 features that represent a model of the gazetteers.

4.2 Integration 2: Gazetteer Features as Input for the CRF Classifier

As NER is a classification task, the system uses features extracted by the BLSTM layer to classify the tokens as one of the possible entity types. CRF is the probabilistic model adopted by NeuroNLP2 to classify a token with an entity type based on the features extracted. Providing the information of the gazetteer to this layer should help the model to better classify tokens. So this integration technique uses the gazetteer features as an additional dimension by concatenating them with the features extracted by the BLSTM.

CoNLL-2003				
	tokens	types	entities	sentences
Train	204567	23624	23499	14987
Dev	51578	9967	5942	3466
Test	46666	9489	5648	3684
DPD				
	tokens	types	entities	sentences
Train	4748	636	1757	450
Dev	296	138	122	49
Test	2315	379	583	200

Table 2: Statistics about data sets used for our experiments.

	dev ∩ train	test ∩ train	test ∩ gazetteers
CoNLL-2003	50%	35%	35%
DPD	48%	26%	33%

Table 3: Unique entities overlap between various sets. The percentage refers to the count of unique entities in the first dataset.

An example of this integration methodology applied to the features provided by the NN$_g$ classifier is presented in Figure 3. It is important to notice that, like in the previous integration approach, NN$_g$ is pre-trained and it is not jointly trained with NeuroNLP2.

5 Data Sets

In this section we describe the two datasets and the various gazetteers used for our experiments. The first dataset is CoNLL-2003 (Tjong Kim Sang and De Meulder, 2003), for named entity recognition, while the second one is a novel dataset (DPD) (Magnini et al., 2018), specifically focused on nominal entities. In Table 2 we report the main characteristics of the two datasets and the partitions used for the experiments, while in Table 3 we report the intersection among entities in the various sets (e.g. how many entities in the test set can be also found in the training set or the gazetteers). These percentages are a rough indicator of: (i) how a perfect match baseline using gazetteers can perform, and (ii) how much the NeuroNLP2 system can take advantage of already seen entities during training phase.

We describe the two datasets with more detail in the following two paragraphs.

CoNLL-2003 is a dataset specifically devoted to named entities: persons, locations, organizations and names of miscellaneous entities that do not belong to the previous three groups. While the data set comes in two languages (English and German) in this work we focus on English, whose

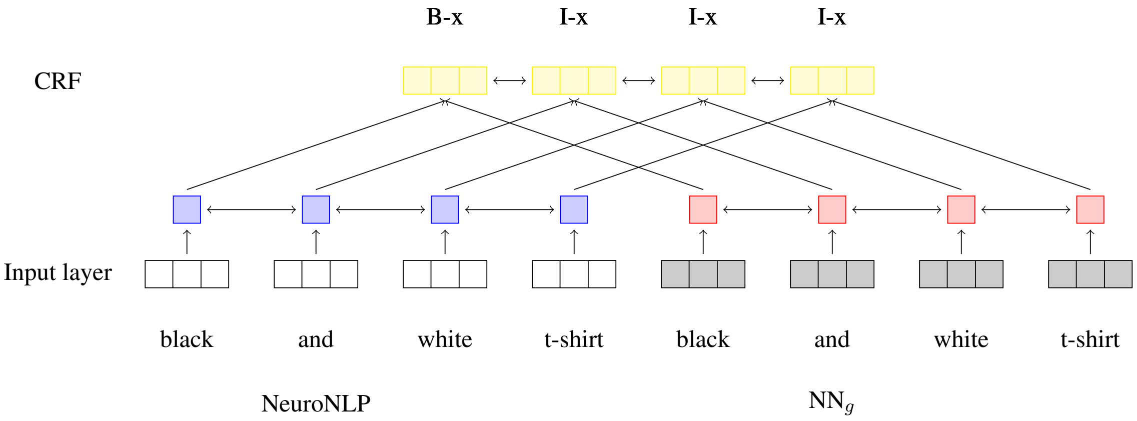

Figure 3: NeuroNLP + NN$_g$ - the output layer of the two systems is combined into the CRF layer. For NN$_g$, a fully connected layer with 32 nodes and a ReLU has been substituted for the original SoftMax layer (in red).

data was taken from the Reuters news corpus. For instance, the sentence "EU rejects German call to boycott British lamb" is tagged as follows in CoNLL-2003:

<EU>$_{ORG}$ rejects <German>$_{MISC}$ call to boycott <British>$_{MISC}$ lamb.

DPD – Diabetic Patients Diary – is a data set in Italian made of diary entries of diabetic patients. Each day the patient has to write down what s/he ate in order to keep track of his/her dietary behavior. In this data set, which is much smaller than CoNNL-2003, all entities of type `FOOD` have been manually annotated by two annotators (inter-annotator agreement is 96.75 dice coefficient). Sentences in the dataset have a telegraphic style, e.g. the main verb is often missing, resulting in a list of foods like the following:

"<risotto ai multicereali e zucchine>$_{FOOD}$ <insalata>$_{FOOD}$ e <pomodori>$_{FOOD}$" ("<risotto with multigrain and zucchini> <salad> and <tomatoes>").

Entity Gazetteers. In Table 4 we describe the gazetteers that we have used in our experiments for the two datasets, reporting, for each entity type, sizes in terms of number of entity names, the average length of the names (in number of tokens), plus the length variability of such names (standard deviation). We also report additional metrics that try to grasp the complexity of entities names in the gazetteer: (i) the normalized type-token ratio (TTR), as a rough measure of how much lexical diversity is in the nominal entities in a gazetteer, see (Richards, 1987); (ii) the ratio of type1 tokens, i.e. tokens that can appear in the first position of an entity name but also in other positions, and type2 tokens, i.e. tokens appearing at the end and elsewhere; (iii) the ratio of entities that contain another entity as sub-part of their name. With these measures we are able to partially quantify how difficult it is to recognize the length of an entity (SD), how difficult it is to individuate the boundaries of an entity (ratio of type1 and type2 tokens), how much compositionality there is starting from basic entities (i.e. how many new entities can be potentially constructed by adding new tokens - sub-entity ratio). Note that type1 and type2 ratios can cover some cases in common with sub-entity ratio, but they model different phenomena: given *white t-shirt*, the entity name *black and white skirt* represents a case of type1 token for *white* but without sub-entity matching, while *white t-shirt with long sleeves* represents a sub-entity matching without making *white* a type1 token.

6 Experiments

The experimental results for the various approaches that use gazetteers as features in the context of a neural entity recognition system, are discussed in this Section. For all experiments, the hyper-parameters of the neural model for both NN$_g$ and NeuroNLP2 are the same as in (Guerini et al., 2018) and (Ma and Hovy, 2016) respectively.

6.1 Overall Results

Tables 5 and 6 show the results of gazetteer integration as embedding and as CRF features, respectively. The NeuroNLP2 model benefits significantly from the gazetteer representation of NN$_g$, especially for the DPD dataset (with an increment of 2.54 in terms of F1). The combination of Neu-

Data Set	Gaz.	#entities	#tokens	length ± SD	TTR	type1(%)	type2(%)	sub-entity(%)
CoNNL	PER	3613	6454	1.79 ±0.54	0.96	19.00	04.63	23.60
	LOC	1331	1720	1.29 ±0.69	0.97	04.66	04.33	10.14
	ORG	2401	4659	1.94 ±1.16	0.91	09.35	15.06	19.44
	MISC	869	1422	1.64 ±0.94	0.89	08.61	08.73	19.85
DPD	FOOD	23472	83264	3.55 ±1.87	0.75	17.22	22.97	11.27

Table 4: Gazetteers used in the experiments. Description is provided in terms of number of entity names, total number of tokens, average length and standard deviation (SD) of entities, type-token ratio (norm obtained by repeated sampling of 200 tokens), type1 and type2 unique tokens ratio and sub-entity ratio.

	CoNLL				DPD			
	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
NeuroNLP2	98.06	91.42	90.95	91.19	88.47	77.17	74.79	75.96
NeuroNLP2 + single token	98.06	91.53	90.51	91.02	88.29	75.63	77.19	76.40
NeuroNLP2 + multi token	98.08	91.41	90.76	91.08	88.98	78.90	76.33	77.59
NeuroNLP2 + NN$_g$	98.05	91.41	91.02	**91.22**	89.89	79.68	77.36	**78.50**

Table 5: Experimental results using gazetteers as features together with embeddings.

roNLP2 and NN$_g$ reaches state-of-art performance on ConNLL-2003 when it is added as embedding feature, while both the single token and the multi-token approaches do not improve the overall results. It can also be clearly seen that providing the gazetteer feature to CRF is a deteriorating choice, as the model probably tends to over-fit to the gazetteer information resulting in a drop of performance. On the other hand, using gazetteer features as part of embedding dimensions helps the model to adapt better when the training data are very few, like in the DPD dataset. Furthermore, the results on the DPD dataset of NeuroNLP2 + NN$_g$, compared to the others, show that NN$_g$ correctly generalizes nominal entities from the gazetteer, improving both Recall and Precision with respect to the multi-token approach.

6.2 Impact of Training Size

Neural network architectures are data-hungry models, requiring large amounts of training data in order to generalize. In those scenarios where the amount of available training is not an issue the effect of the gazetteer on the model performance is negligible, as the model learns to generalize on the large number of annotated sentences. However, for domains where there is scarcity of training data, the gazetteer feature is much appreciated for a better performance. To understand this effect, we simulated a low data scenario for the CONLL-2003 dataset by training the model on a small amount of data. The test and dev datasets are kept the same, while varying only the training data size. Tables 7 and 8 show the performance of the models with varying training data sizes on CONLL and DPD datasets, respectively. We can infer that for Named Entity Recognition using token presence is not the right approach especially when the gazetteer is well formed and with little noise. The multi-token feature approach is more consistent, and it improves the performance of the NeuroNLP2 model by at least 3 points over all data sizes used. However, these approaches tend to be inconsistent when learning nominal entities. Results show that NN$_g$ proves to be more robust for nominal entities and provides a more consistent performance indicating its impact in recognizing compositional entities. We can see that, for nominal entities, the single token approach is not recommended, as both the learning curve and the gazetteer size effect show (see Table 9).

Results are variable particularly for food names, which are nominal entities that can considerably change in length, contain stop-words, numbers or nouns that usually can appear in other contexts (i.e. *Miami beach* cocktail, *chinese* chicken, *white* wine, *pad* thai, *energy* balls...).

6.3 Impact of Gazetteer Size

In this section we investigate the impact of reducing the number of entities in the gazetteer, based on a "less common" principle, i.e. removing rare, but numerous, entries. As we can see in Table 9, the single token approach does not work well on food nominal entities, giving variable and unreliable results. In addition, removing entries that

	CoNLL				DPD			
	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
NeuroNLP2	98.06	91.42	90.95	91.19	88.47	77.17	74.79	75.96
NeuroNLP2 + single token	94.82	86.90	71.03	78.17	85.75	69.79	68.95	69.37
NeuroNLP2 + multi token	92.20	85.44	59.24	69.97	87.52	74.23	74.61	74.42
NeuroNLP2 + NN$_g$	97.96	90.95	90.53	90.74	89.37	78.56	74.79	**76.63**

Table 6: Experimental results using gazetteers as features for CRF.

	Data Size			
	100	200	300	450
NeuroNLP2	47.31	56.78	56.96	69.55
NeuroNLP2 + single token	47.70	59.25	61.32	71.66
NeuroNLP2 + multi token	**51.24**	**62.77**	**63.19**	**74.63**
NeuroNLP2 + NN$_g$	42.30	57.99	58.65	69.20

Table 7: Learning Curve on the CONLL 2003 dataset. Columns report the number of sentences in the training dataset.

	Data Size			
	100	200	300	450
NeuroNLP2	55.99	**74.93**	73.88	75.96
NeuroNLP2 + single token	48.93	72.29	**78.53**	76.40
NeuroNLP2 + multi token	**63.82**	72.80	77.30	77.59
NeuroNLP2 + NN$_g$	52.85	72.24	77.02	**78.50**

Table 8: Learning Curve on the DPD dataset. Columns report the number of sentences in the training dataset.

contain tokens that appear less than 10 times helps NN$_g$ to better generalize food names, without focusing on rare and uncommon entities. With this approach the size of the gazetteer is nearly halved, and it is noticeable that 5019 out of 8420 unique tokens in the gazetteer appear only once. Removing common tokens, i.e. those that appear between 10 and 49 times, no model seems to give decent results, with performance lower than the NeuroNLP2 model alone. A last reduction experiment, i.e. removing entities that contain very common tokens, occurring more than 150 times, bringing the gazetteer to a size of only 779 entries, leave NN$_g$ with too few compositions to learn how to generalize, but permits the multi-token approach to give core information to NeuroNLP2, increasing its baseline performance.

	Data Size			
	776 T$_c \geqslant$ 150	4366 T$_c \geqslant$ 50	12477 T$_c \geqslant$ 10	23472 all
NeuroNLP2 + single token	75.90	75.19	70.99	76.40
NeuroNLP2 + multi token	**76.11**	75.05	76.96	77.59
NeuroNLP2 + NN$_g$	74.63	**75.62**	**79.69**	**78.50**

Table 9: Results of reducing gazetteer size on a *less common* principle; in the columns, the first number is the gazetteer size, while the second element represents the minimum number of occurrences for the tokens in the FOOD gazetteer.

7 Related Work

Although, at least to the best of our knowledge, there is no much work specifically addressing the use of gazetteers in nthe context of neural architectures, still there is a number of related contributions which we discuss in this section.

The use of gazetteers with neural networks has been proposed by (Park et al., 2017), who present a neural network model augmented with syllable embedding vectors, parts-of-speech probability vectors, and gazetteer vectors as input features. Although the proposed model showed good performance, there is no attempt to isolate the impact of gazetteers, which is the goal of our work.

A related approach is presented in (Zhao et al., 2017), which ranked first in English NERC evaluation at KBP 2017. Basically this is an extension of (Lample et al., 2016) that includes entity embedding from gazetteers, where embeddings are derived from a noisy gazetteer created using Wikipedia's articles. The gazetteer is derived from the word-entity statistics from (Park et al., 2017).

A good example of work that builds on the idea of creating a statistical model of named entity starting from gazetteers, is presented in (Al-Olimat et al., 2017). The paper focuses on extracting location names from informal and unstructured texts by identifying referent boundaries. The core of the approach is a statistical language model consisting of a probability distribution over sequences of words (collocations) that represent location names in gazetteers. The algorithm uses the relative likelihood of an observed word sequence to decide the boundaries of a location name in tweets. This is similar in spirit to the NN$_g$ used in our approach, although we make use of a neural model rather than a statistical model.

A second work that it is worth to mention is (Yang et al., 2016), which addresses the problem of using gazetteers when training a neural network with few data. In fact, particularly for massive data scenarios like NER on Twitter, collecting a large amount of high quality gazetteers can alleviate the problem of training data scarcity. The paper shows that large gazetteers may cause a side-effect called ‘feature under-training’, i.e. gazetteer features overwhelm the training data and may degrade performance. To solve this problem, the authors propose a dropout conditional random fields, which decreases the influence of gazetteer features with a high weight.

8 Conclusions and Future Work

In this paper we were interested to investigate several options about the use of gazetteers in neural architectures for entity recognition. We conducted several experiments on both named entities (CONLL 2003 - English) and nominal entities (food names in DPD - Italian) and showed that: (i) gazetteer features that are extracted by a separately trained the NN_g classifier are more significant than conventional features based on presence-absence of tokens; (ii) integrating such features as extension of the input embedding outperforms integration at the CRF level; (iii) these findings are particularly significant when either the size of the training data or of the gazetteers are reduced.

As a general comment on the use of gazetteers for neural NER, our experiments highlight that gazetteers are much more useful for nominal entities (e.g. food names) than for named entities (e.g. person names). In this respect, the paper shows that the NN_g approach significantly helps to identifying compositional variants of nominal entities.

In the paper we have based our experiments on the neural model described in (Ma and Hovy, 2016). However, very recently, new models (e.g. BERT (Devlin et al., 2018) and ELMO (Peters et al., 2018)) have been proposed, which have further improved performance on Named Entity Recognition. Based on such expectations, we run a preliminary experiment using the multilingual BERT model on the DPD dataset: results, probably due to the poor performance of the model on Italian food, are still significantly lower than NeuroNLP2, and additional work seems to be necessary to properly take advantage of the full capacity of the BERT model.

Finally, as for future work, we intend to apply the current models to a larger number of scenarios, including utterance understanding for conversational agents.

References

Hussein S Al-Olimat, Krishnaprasad Thirunarayan, Valerie Shalin, and Amit Sheth. 2017. Location name extraction from targeted text streams using gazetteer-based statistical language models. *arXiv preprint arXiv:1708.03105*.

Silvana Marianela Bernaola Biggio, Manuela Speranza, and Roberto Zanoli. 2010. Entity mention detection using a combination of redundancy-driven classifiers. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Jason PC Chiu and Eric Nichols. 2015. Named entity recognition with bidirectional LSTM-CNNs. *arXiv preprint arXiv:1511.08308*.

Jinho D Choi. 2016. Dynamic feature induction: The last gist to the state-of-the-art. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 271–281.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.

George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. 2004. The automatic content extraction (ace) program - tasks, data, and evaluation. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC-2004)*, Lisbon, Portugal. European Language Resources Association (ELRA). ACL Anthology Identifier: L04-1011.

Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A Smith. 2015. Transition-based dependency parsing with stack long short-term memory. *arXiv preprint arXiv:1505.08075*.

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

Marco Guerini, Simone Magnolini, Vevake Balaraman, and Bernardo Magnini. 2018. Toward zero-shot entity recognition in task-oriented conversational agents. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 317–326.

Luheng He, Kenton Lee, Mike Lewis, and Luke Zettlemoyer. 2017. Deep semantic role labeling: What works and whats next. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 473–483.

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

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. *CoRR*, abs/1603.01360.

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)*, volume 1, pages 1064–1074.

Bernardo Magnini, Vevake Balaraman, Mauro Dragoni, Marco Guerini, Simone Magnolini, and Valerio Piccioni. 2018. Ch1: A conversational system to calculate carbohydrates in a meal. In *International Conference of the Italian Association for Artificial Intelligence*, pages 110–122. Springer.

Geonwoo Park, Hyeon-Gu Lee, and Harksoo Kim. 2017. Named entity recognition model based on neural networks using parts of speech probability and gazetteer features. *Advanced Science Letters*, 23(10):9530–9533.

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

Lance A. Ramshaw and Mitchell P. Marcus. 1995. Text chunking using transformation-based learning. *CoRR*, cmp-lg/9505040.

Brian Richards. 1987. Type/token ratios: What do they really tell us? *Journal of child language*, 14(2):201–209.

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-Volume 4*, pages 142–147. 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.

Huasha Zhao, Yi Yang, Qiong Zhang, and Luo Si. 2017. Improve neural mention detection and classification via enforced training and inference consistency. *Proc. TAC2017*.