Toward Zero-shot Entity Recognition in Task-oriented Conversational Agents

Marco Guerini[1], **Simone Magnolini**[1,2], **Vevake Balaraman**[1], **Bernardo Magnini**[1]
[1] Fondazione Bruno Kessler, Via Sommarive 18, Povo, Trento — Italy
[2] AdeptMind Scholar
{guerini,magnolini,balaraman,magnini}@fbk.eu

Abstract

We present a domain portable zero-shot learning approach for entity recognition in task-oriented conversational agents, which does not assume any annotated sentences at training time. Rather, we derive a neural model of the entity names based only on available gazetteers, and then apply the model to recognize new entities in the context of user utterances. In order to evaluate our working hypothesis we focus on nominal entities that are largely used in e-commerce to name products. Through a set of experiments in two languages (English and Italian) and three different domains (furniture, food, clothing), we show that the neural gazetteer-based approach outperforms several competitive baselines, with minimal requirements of linguistic features.

1 Introduction

In this paper we focus on user utterance understanding, where a conversational system has to interpret the content of a user dialogue turn. At this step, most of conversational systems try to capture both the intent of the utterance and the relevant entities and relations that are mentioned. As an example, given a user query like: *Can I find a Canada Goose parka blue for -30?*, an online shop assistant should be able to recognize that the intent of the utterance is 'Search' and that the following entities are mentioned: Product_Category = parka; Brand = Canada Goose; Color = blue; Min_temperature = -30. We are particularly interested in application domains, like e-commerce, which show specific characteristics: large variety of entity names for the same category (e.g. *a black and white t-shirt*, *black pants*, *white vintage shoes* are all names of clothes); compositionality of entity names (e.g. *black pants*, *black short pants*); utterances with multiple occurrences of the same entity category (e.g. "I would like to order a *salami pizza* and two *mozzarella cheese sandwiches*" contains two occurrences of food); strong requirements of multilinguality (e.g. *scarpe bianche vintage* and *white vintage shoes*). Finally, we are interested in domains where available repositories can only cover a portion of the possible entity names that a user can express in an interaction.

Our working hypothesis is that, in such scenarios, current entity recognition approaches based on supervision (i.e. we call them *pattern-based* as they need utterances annotated with entities in the context they occur), need a huge amount of supervision to manage the variety of entity names, which would make those approaches ineffective in most practical situations. Thus, we propose an entity recognition method, we call it *gazetteer-based*, which takes advantage of available entity names for a certain category to train a neural model that is then applied to label new unseen entities in a user utterance. This method shares several features with recent proposals in zero-shot learning (Xie et al., 2016), as we do not assume any annotated utterances at training time, and we make use of entity names as "side information".

We run several experiments on three e-commerce domains (furniture, food, clothing) and two languages (English and Italian), with different characteristics in terms of entity names, and show that: (i) the gazetteer-based approach significantly outperforms the pattern-based approach in our domains and languages; (ii) the method captures linguistic properties of the entity names related to their compositionality, which are reliable indicators of the complexity of the task.

The paper is structured as follows. Section 2 introduces the entity recognition task we are

Proceedings of the SIGDIAL 2018 Conference, pages 317–326,
Melbourne, Australia, 12-14 July 2018.

addressing. Section 3 provides background and relevant related work. Section 4 describes the gazetteer-based methodology that we adopt for entity recognition in user utterances. Finally, section 5 and 6 describe, respectively, the experimental setting and the obtained results.

2 Entity Recognition for E-commerce

Common conversational systems adopt a slot filling approach as semantic representation of the utterance content. Usually, it is assumed that the utterance contains just one entity for each slot. In addition, typical entities corresponds to named entities (e.g. locations) or to almost closed classes (e.g. time, dates, quantities, currencies). Although this is substantially true for several popular task oriented scenarios, like flight booking (a well known dataset is ATIS – Air Travel Information Services), point of interest navigation, and calendar scheduling (for instance the dataset used in (Eric and Manning, 2017)), other conversational scenarios show different characteristics. In this section we focus on conversational agents for the e-commerce scenario, and highlight the characteristics which we believe are relevant for entity recognition.

Task-oriented dialogue. E-commerce chat-bots are supposed to carry on a task-oriented dialogue whose goal is helping the user to select products presented in an online shop, and, ultimately, buy them. For the purposes of this paper we restrict our attention to written chat-style dialogues (i.e. voice is not considered).

Entity names. The main focus of the interaction is on products (i.e. users search, compare, assess information on products they are interested in). Products can be referred to in several ways, including their descriptions (e.g. *a round table with a marble top*), proper names (e.g. *Adidas Gazelle*), or with a mix of them (e.g. *a white Billy shelf*). Depending on the complexity of the domain, a single online shop may manage from thousands to several hundreds of thousand of different products, with hundreds of variants (e.g. size and colour for clothes). Throughout this paper, we refer to such product descriptions as *entity names*. As we will see, there is a high variance in the way online vendors assign and manage such names. For the purposes of this paper, it is relevant to notice that taking advantage of e-commerce website catalogs, it is relatively easy to download repositories of entity names for a large variety of products, and for several languages. On the other hand, a structured description of such entities - in term of slot-value pairs - is often missing. We call these repositories of entity names *gazetteers*.

Conversational patterns. Conversational patterns in e-commerce dialogues are relatively simple. High level user intents vary from searching for one or more products, asking to compare characteristics of products, and finalizing the purchase. Although there are just a few datasets available to support our intuition (e.g. the Frames dataset presented in (El Asri et al., 2017)), we may assume that the context in which product names appear is quite limited. Compared to other scenarios (e.g. booking hotels and flights), it is quite frequent that user mention more than one product in the same utterance (e.g. "Please deliver at home a *salami pizza*, a *pepperoni pizza with onions* and two *mozzarella cheese sandwiches*").

Multilinguality. E-commerce is becoming more and more multilingual. The market is worldwide and vendors offer navigation in several languages. For our purposes a strong requirement is that approaches for entity recognition must be easily portable through languages.

3 Background and Related Work

In this section we report useful context for the *gazetteer based* approach that will be described in Section 4. We focus on entity recognition, zero-shot learning and generation of synthetic data.

3.1 Entity Recognition

Entity recognition has been largely approached as a sequence labeling task (see, for instance, the Conll shared tasks on named entities recognition (Tjong Kim Sang and De Meulder, 2003)). 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 utterance "I would like to order a salami pizza and two mozzarella cheese sandwiches", would be labeled as shown in Table 1.

We refer to the Automatic Content Extraction program - ACE (Doddington et al., 2004), where

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.

two main entity classes are distinguished: named entities and nominal entities. We focus on the latter, as this is more relevant for utterance understanding in the e-commerce scenario. Nominal entities are noun phrase expressions describing an entity. They can be composed by a single name (e.g. *pasta*, *carpet*, *parka*) or by more than one token (e.g. *capri sofa bed beige*, *red jeans skinny fit*, *lightweigh full frame camera*, *grilled pork belly tacos*). Nominal entities are typically compositional, as they do allow morphological and syntactic variations (e.g. for food 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 addition to adjectival and prepositional modifiers, conjunctions are also very frequent (e.g. *beef and bean burritos*, *black and white t-shirt*). Compositionality is crucial in our approach, as we take advantage of it to syntethically generate negative training examples for a certain entity category, as detailed in Section 4.1.

3.2 Zero-shot Learning

In conversational agents there is a general lack of data, both annotated and unannotated, as real conversations are still not widely available for different domains and languages. To overcome this limit, in our gazetteer-based approach we take advantage of the fact that it is relatively easy to obtain repositories of entity names for several categories (e.g. food names, locations, movie titles, names of products, etc.). We use such repositories as "side information" in zero-shot learning to recognize entity names for a certain class, even if no annotated utterances are available for that class. While similar approaches have been already proposed to improve portability across domains (e.g. (Bapna et al., 2017) uses slot names as side information), in this paper we take advantage of the zero-shot approach focusing on large repositories of compositional entity names.

Several approaches have been proposed to implement zero-shot learning, including those that use multiple embeddings (Norouzi et al., 2013), those that extract features that generalize through different domains (Socher et al., 2013), and those that recast zero-shot learning as a domain adaptation problem (Elhoseiny et al., 2013).

3.3 Synthetic Data Generation

Partly due to the need of large amounts of training data to feed neural networks, recently there has been a diffused interest on methods for automatically generate synthetic data (see (Jaderberg et al., 2014)). The effectiveness of synthetic data generation has been shown in several domains, including the generation of textual descriptions of visual scenes (Hoag, 2008), and of parallel corpora for Machine Translation (Abdul-Rauf et al., 2016). Alternative approaches to data generation for conversational agents are based on simulated conversations (Shah et al., 2018). As for the e-commerce domain, because of the dramatic scarcity of available datasets, we were forced to use synthetic generation in two cases: negative training examples for entity names, used to train our gazetteer-based approach, and lexicalization of utterances, used for testing the performance of our approach.

4 NN_g Entity Recognition

In our zero-shot learning assumption we propose a neural gazetteer-based approach, which includes two main components: a neural classifier (NN_g) trained solely on the entity names in a gazetteer, described in Section 4.1, and the entity tagger that applies the neural classifier to a user utterance, described in Section 4.2.

4.1 NN_g Classifier

The NN_g classifier is the core of the gazetteer-based approach. It is implemented using a multilayer bidirectional LSTM (Schuster and Paliwal, 1997) that classifies an input sequence of tokens either as entity or non-entity for a certain entity category, with a certain degree of confidence. We base our NN_g classifier on the system proposed in (Lample et al., 2016), which was modified to match the peculiarities of the gazetteer-based approach: (i) we extend it as a 3-layer biLSTM with 120 units per layer and a single dropout layer

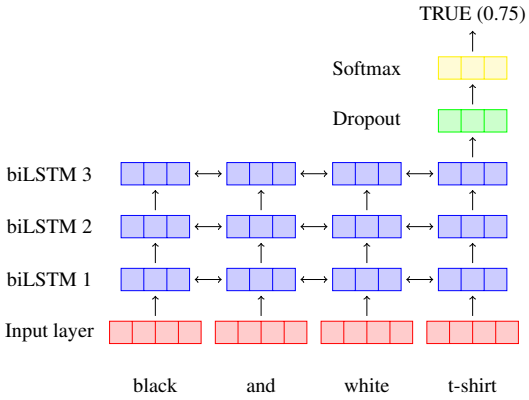

Figure 1: Structure of the Neural Gazetteer (NN_g) entity classifier. The input layer concatenates the features in a single vector.

(dropout probability of 0.5) between the third biLSTM and the output layer. This topology (see Figure 1) has been empirically defined using the train and dev portions of the synthetic gazetteers described in section 5.2. (ii) The output layer is a softmax layer – instead of a CRF layer – because the goal of NN_g is to classify the whole sequence and not to tag each single token using the IOB format. The softmax layer provides the probability of a sequence being positive or negative for a certain category, based on the output from the previous layers. We use this probability as a *confidence score* for a sequence being positive or negative.

This multilayer biLSTM is meant to build an internal representation of the core compositional structure of the entity names that are listed in the gazetteer, and to generalize such structure to recognize new entity names of the same category.

Synthetic Training Data. In order to train the NN_g classifier, we need not only positive examples (i.e. entity name), but also negative ones, i.e. sub-sequences of an utterance where no entities are present or where only parts of the entity name are present. To obtain such negative examples we used the following methodology based on synthetic generation. For each entity name i in a gazetteer G, negative counterparts can be obtained either using a sub-sequence of i (making sure it is not present in the gazetteer), or by taking i and adding tokens at the beginning or end of it (or both), following the pattern $t_1 + i + t_2$, where t_1 is the ending token of a random entity in G and t_2 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, t_1 and t_2 can be tokens randomly extracted from a generic corpus, so as to mimic cases when the entity is expressed in isolation. For example, if the initial positive example is *black and white t-shirt*, the possible negative sub-sequences that are generate are: | *black* | *white* | *black and* | *and white* | *black and white* |. The sub-sequences | *white t-shirt* | *t-shirt* | are not considered because they are already included in the gazetteer as positive examples. Adding tokens, using the pattern $t_1 + i + t_2$, we obtain other potential negative examples: | *buy black and white t-shirt* | *black and white t-shirt and sweater* | *buy black and white t-shirt and sweater* |, and so on. According to this procedure, we generate more negative examples than positive. In order to avoid an unbalanced dataset, we randomly select two negative examples per positive one: a sub-sequence and an example surrounded by other words, resulting in a 1:2 proportion.

Classifier Features. The NN_g classifier combines several features: two different word embeddings (i.e. generic and specific), a char-based embedding, and seven handcrafted features. The generic word embedding is employed to capture generic language use, and it is similar to the one used in (Lample et al., 2016). For English it was trained using the English Gigaword version 5, while for Italian it was trained using a dump of the Italian Wikipedia. We use an embedding dimension of 64 for both English and Italian, a minimum word frequency cutoff of 4, and a window size of 8. The second word embedding is employed to capture language use that is specific for each domain, and it is extracted using the training gazetteer as corpus, with a dimension of 30, a minimum word frequency cutoff of 1, and a window size of 2. Finally, the char-based embedding with a dimension of 50 is still based on (Lample et al., 2016) and it is trained on the domain gazetteers. Its function is to deal with out of vocabulary terms and possible misspellings.

Handcrafted features are meant to explicitly represent the core structure of a typical entity name. We consider seven features of 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

name containing a certain token; (v) the average position of the token in the entity name it appears in; (vi) the bigram probability with reference to the previous token in the entity name; (vii) the list of all the possible PoS associated to the token.

4.2 NN$_g$ Tagger

The neural classifier described in the previous section is applied to all the sub-sequences of a certain utterance (see algorithm 1), in order to select candidates entity names for a certain category. After classification the algorithm takes a further step to select the actual entities, by ranking the candidates according to the confidence score provided by the classifier, and by selecting the top not overlapping candidates. As an example, the utterance "I'm looking for golden yellow shorts and dark blue shirt" contains six sub-sequences that are classified as positive by the NN$_g$ classifier (lines [1-5]): | *shorts* | *yellow shorts* | *golden yellow shorts* | *shirt* | *blue shirt* | *dark blue shirt* |, while all other sub-sequences, such as: | *I'm looking* | *looking for a golden* | *shorts and dark* | *dark blue* |, are classified as negative. Then, positive examples are ranked according to their confidence score (lines [6]): | *golden yellow shorts* | *yellow shorts* | *dark blue shirt* | etc. Finally, *golden yellow shorts* is selected while *yellow shorts* is discarded because the latter overlaps with the former. Likewise *dark blue shirt* is selected since it is not overlapping with other already selected sub-sequences while all remaining ones are discarded (lines [7-11]).

Algorithm 1 NN$_g$ Tagger

```
1:  for sub-sequence in utterance do
2:      if sub-sequence is an entity then
3:          add sub-sequence to entity-list
4:      else
5:          discard sub-sequence
6:  order entity-list by confidence-score
7:  for element in entity-list do
8:      if element not overlap previous elements
    then
9:          tag element as entity
10:     else
11:         discard element
```

5 Experimental Setting

In this section we first introduce two alternative approaches for entity recognition that we used as comparison with NN$_g$, and then the datasets that are used for our experiments.

Algorithm 2 Rule-based entity recognition

```
1:  G : tokens in Gazetteer - excluding stopwords.
2:  morpho : morphological variations of token.
3:  POS : possible PoS tags for the token.
4:  bigram : All bi-grams in Gazetteer.
5:
6:  for token in utterance do
7:      if token is in an NP chunk then
8:          if IN_GAZETTEER(token) then
9:              tag token as entity
10:     else
11:         if any(morpho[word] in G) then
12:             if any(PoS[word] is noun) then
13:                 tag token as entity
14: for token_i in utterance do
15:     if bigram(token_i, token_{i+1}) exits then
16:         tag token_i and token_{i+1} as entity
17: Format tags to IOB notation
```

5.1 Entity Recognition Algorithms

We have compared the NN$_g$ approach described in Section 4 with two alternative entity recognition approaches: an unsupervised rule-based algorithm, which takes advantage of both the entity gazetteer and of linguistic information about chunking, and a supervised algorithm that needs annotated sentences as training.

Rule-based entity recognition. This approach is based on (Eftimov et al., 2017), a system that uses a terminological-driven and rule-based named entity recognizer, taking advantage of both entity dictionaries and rules based on chunks. The core strategy is that a chunk in a text is recognized as belonging to a category C if any of its tokens are present in the gazetteer for category C. The approach in (Eftimov et al., 2017) is tailored to a single domain/language and involves merging successive chunks into a single one based on the rules imposed by the algorithm. We extended the approach by adding morphological features and the possible PoS of a word, for which we used TextPro (Pianta et al., 2008), (see Algorithm 2).

We assume that the dictionary+chunk algorithm is particularly suitable for compositional entities. In fact, actual entities in a text can still be recognized even if the perfect match is not present in the original dictionary. For example, the tar-

get entity *white t-shirt with long sleeves* can be correctly identified as long as there are entities in the gazetteer that contain the tokens of interest, such as *black and white t-shirt* and *red t-shirt with long sleeves*.

Neural pattern-based entity recognition (NN_p). We used the bidirectional LSTM architecture introduced by (Lample et al., 2016) for named entity recognition. Given an input embedding for a token in the utterance, the outputs from the forward and backward LSTM are concatenated to yield the context vector for the token, which is then used by a CRF layer to classify it to the output type (O, I-, B-). There are 100 LSTM units and a dropout of 0.5 is applied to the BiLSTM layer. To train the NN_p model, we used pre-trained embeddings on Wikipedia corpora. This helps the model to adapt itself to unseen words in the test data, provided they have an embedding.

As expected, the proposed NN_p model is highly efficient to identify the context in which an entity occurs in the utterance. However, it is also prone to make errors in the sequence of the tags (i.e. tagging a token to be I- without a preceding B- tag). This is because, when trained with limited data, the entities in the training data do not cover all possible tags for a token, and also not all the possible entities (Lample's model was trained on more than ten thousand sentences per language, but in our scenario the training data is limited to few hundred sentences). For this reason, and to highlight the model's capability to identify the context of an entity, at test time the outputs of the model are post-processed to comply with the IOB notation; e.g. tag sequences such as O, I-, B-, I- are modified to O, B-, I-, I-.

5.2 Datasets

We experimented entity recognition in three e-commerce domains and two languages for a total of six configurations. The three domains are respectively: *food*, *clothing* and *furniture*. Languages are Italian and English. In order to run our experiments the following datasets were used.

Entity gazetteers (positive examples for NN_g). We collected a gazetteer of nominal entities for each domain-language pair. To allow for consistent comparisons across languages and domains we scraped just one website per domain and extracted the English/Italian gazetteers versions. In Table 2 we describe each gazetteer, reporting its size 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, SD). We also report additional metrics that try to grasp the complexity of entity name in the gazetteer: (i) the normalized type-token ratio (TTR), as a rough measure of how much lexical diversity there is for the nominal entities in a gazetteer, see (Richards, 1987); (ii) the ratio of $type_1$ tokens, i.e. tokens that can appear in the first position of an entity name but also in other positions, and $type_2$ 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, how difficult is to individuate the boundaries of an entity (ratio of $type_1$ and $type_2$ tokens), how much compositionality there is starting from basic entities (i.e. how many new entities can be potentially constructed by adding new tokens). Note that $type_1$ and $type_2$ ratios can cover 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 $type_1$ token for *white* but without sub-entity matching, while *white t-shirt with long sleeves* represents a sub-entity matching without making *white* a $type_1$ token.

Synthetic Gazetteers (positive + negative examples for NN_g) (SG). To train NN_g, we apply the methodology described in Section 4.1 to obtain synthetic negative data. After splitting each gazetteer using a 64:16:20 ratio (train:dev:test), we created the aforementioned data sets, where – for each entity *i* (positive example) present in the train-dev splits – we added two negative examples obtained by randomly selecting one of the methodologies described in Section 4.1. The optimal number of negative examples was obtained during the training phase by varying their ratio.

Synthetic Utterances (training for NN_p, test data for all approaches) (SU). To test our approaches we used synthetic sentences produced by lexicalizing templates, following the idea presented in (Cheri and Bhattacharyya, 2017; He et al., 2017). These recent approaches show the feasibility of using synthetic sentences both for training and test. More generally, there's a growing interest in using synthetic data for conversational agents, e.g. the *bAbI* datasets - meant to de-

Gazetteer	#entities	#tokens	length ± SD	TTR	type$_1$(%)	type$_2$(%)	sub-entity(%)
food_EN	58539	265726	4.54 ±2.53	0.76	21.37	14.61	10.70
food_IT	29340	101860	3.47 ±1.80	0.69	16.90	22.44	13.31
furniture_EN	3595	13601	3.78 ±1.48	0.62	3.24	7.10	2.75
furniture_IT	2624	10045	3.83 ±1.56	0.63	2.32	7.61	3.43
clothing_EN	36290	127944	3.53 ±1.05	0.63	13.12	0.30	12.60
clothing_IT	34698	130106	3.75 ±1.24	0.64	0.29	14.71	13.50

Table 2: Gazetteers used in the experiments. Description in terms of number of entity names, total number of tokens, average length and standard deviation (SD) of entities, type-token ratio (TTR, norm obtained by repeated sampling of 200 tokens), type$_1$ and type$_2$ unique tokens ratio and sub-entity ratio.

Intent	Template
Select	I'm fine with <entity>
Description	Could you explain to me what <entity> is
AddToList	I want to put both <entity> and <entity> on my list
RateItem	I want to give <entity> two stars

Table 3: Examples of intents and corresponding templates used to generate test utterances.

NN$_g$ features config.	F1	SDV
Gazetteer-info	88.08	4.94
Handcrafted	86.39	5.90
Embeddings	87.66	4.10
All	89.95	4.05

Table 4: Average F1 and standard deviation for various features configurations of NN$_g$ over the six SG data sets (three domains and two languages).

velop learning algorithms for text understanding and reasoning - were all constructed in a synthetic way (Weston et al., 2015).

We created 237 templates for English and the same amount for Italian. These templates were manually designed in order to be domain independent (e.g. using terminology that can be applied to any domain), and correspond to typical intents that can be found in the e-commerce scenario (e.g. buy, add to list, rate item, etc.) and were evenly distributed in order to contain 1 to 3 entity names. A few examples are given in Table 3.

We split the templates in a 64:16:20 ratio (train:dev:test) before lexicalization: to lexicalize SU$_{train}$ we randomly choose entities that were in the train split of the gazetteers, while for SU$_{test}$ we randomly choose entities than were in the test split of the gazetteers. It should be noted that we used this procedure to better isolate the effect of entity name and their compositional nature over learning approaches, in fact: (i) we controlled for the impact of patterns on learning by using the same patterns across data sets train and test splits. (ii) we made the task more challenging than in standard situations, since no entity present in the training can be present in the test sets as well. In this way we can assess the ability of the approaches to learn the structure of entity names and generalize it to new examples. So, for example, a simple baseline that uses exact match over the train gazetteers to identify entities in the test sentences would report a F1 of 0.

Finally, according to our zero-shot assumption, the NN$_g$ is trained using solely SG, while its performances are computed using SU$_{test}$.

6 Experiments and Results

We run two different sets of experiments to explore the impact of compositionality on the task of entity recognition. The first set was meant to find the optimal feature configuration for NN$_g$, and the second one was the comparison of the three main approaches over the six SU datasets.

1. Experiments with NN$_g$ on SG. We run a set of experiments to assess the best feature configuration for the gazetteer-based approach. In Table 4 we report the overall results of NN$_g$ using different feature configurations, over the six SG data sets. The topological configuration of NN$_g$ is kept constant, as described in Section 4. As can be seen, the configuration using all features is the best one (F1 89.95), and also the one with the lowest standard deviation (4.05). This means not only that this configuration provides the best results on average but also the most consistent ones across all data sets. Interestingly, the configuration that uses no external linguistic knowledge (Gazetteer-info)

	English			Italian		
	Food	Furniture	Clothing	Food	Furniture	Clothing
Baseline 1: Rule-based	5.74	33.61	34.75	21.26	25.13	44.78
Baseline 2: NN_p	25.53	43.67	61.76	14.79	25.33	22.88
NN_g approach	32.43	63.28	76.92	37.17	40.41	62.64

Table 5: Experimental results (F1) over the six domain-language data sets.

is the second best, indicating that even in the worst case, in which no linguistic resource is available, we can still expect to obtain competitive results.

2. Experiments and Comparison on SU. Table 5 reports the comparison among the rule-based baseline, the NN_p baseline, and the NN_g approach. NN_g is the best approach on all domains and languages. This confirms our initial hypothesis that the structure of entity names induced by gazetteers is fundamental when having little knowledge of the context in which entities occur within utterances (i.e. having few training examples).

It should be noted that the effect of entity name complexity (reported in Table 2) emerges clearly from the experiments: all the approaches tend to be affected by it. In both languages we have the following order in term of performances *food* < *furniture* < *clothing*. While for *food* results are evident (the highest length-SD, TTR, $type_1$ and $type_2$ token ratios and high sub-entity ratio affect the performances even if the gazetteers are big) for *furniture* and *clothing* we need to look closer at the metrics in Table 2. Neglecting the possible effects of gazetteer size, we see that *clothing* tends to have higher ratio of $type_1$ or $type_2$ tokens: this is due to the large use of modifiers, such as colour, typical of the domain (depending on language the modifier is attached before or after the head *white t-shirt* vs *maglietta bianca*). Still, being the other token type almost 0, either the beginning or the end of an entity name is unambiguous, and in case of adjacent entities in a sentence this is enough to recognize the boundaries between the two.

The NN_g version that uses only gazetteer features (i.e. no linguistic knowledge is assumed), even if not reported in Table 5, showed to perform more poorly than the version using all features. Still, it is competitive against NN_p, outperforming it in five SU data sets out of six, and providing an average F1 improvement of 10 points.

Finally, in Table 6 we report the results of an additional analysis, where we computed the F1 scores according to the number of entities present in the test sentences (all domain and languages). As can be seen, NN_g is the least sensitive to the number of entities present in the test sentences (i.e. NN_g is the most consistent in term of performance under all circumstances). This can be explained by the fact that NN_g, being focused on recognizing entities rather than patterns, is less sensitive to cases of contiguous occurrences of entities that can be wrongly segmented by other approaches.

#Entities	Rule-based	NN_p	NN_g
One	27.46	47.39	59.04
Two	35.52	45.29	48.12
Three	22.14	24.43	52.42

Table 6: Results (F1) of the three approaches according to the number of entities in the SU datasets.

7 Conclusions and Future Work

We have provided experimental evidence that zero-shot entity recognition based on gazetteers is highly performing. To our knowledge, this is the first time that a neural model has been applied to capture compositionality of entity names. Due to the scarcity of annotated utterances, the proposed approach is particularly recommendable for its portability through different domains and languages. Our experiments have been tested on synthetic data (i.e. utterances semi-automatically generated starting from a set of conversational patterns) in the context of e-commerce chat-bots, taking advantage of some of the characteristics of the scenario. As for the future, we intend to test the approach on natural utterances (i.e. not synthetically generated).

Acknowledgements

This work has been partially supported by the AdeptMind scholarship, and by the CBF EIT Digital project. The authors thank the anonymous reviewers and Hendrik Buschmeier for their help and suggestions.

References

Sadaf Abdul-Rauf, Holger Schwenk, Patrik Lambert, and Mohammad Nawaz. 2016. Empirical use of information retrieval to build synthetic data for SMT domain adaptation. *IEEE/ACM Trans. Audio, Speech & Language Processing* 24(4):745–754.

Ankur Bapna, Gokhan Tur, Dilek Hakkani-Tur, and Larry Heck. 2017. Towards zero shot frame semantic parsing for domain scaling. In *Interspeech 2017*.

Joe Cheri and Pushpak Bhattacharyya. 2017. Towards harnessing memory networks for coreference resolution. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*. pages 37–42.

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)*. European Language Resources Association (ELRA), Lisbon, Portugal. ACL Anthology Identifier: L04-1011. http://www.lrec-conf.org/proceedings/lrec2004/pdf/5.pdf.

Tome Eftimov, Barbara Koroušić Seljak, and Peter Korošec. 2017. A rule-based named-entity recognition method for knowledge extraction of evidence-based dietary recommendations. *PLoS ONE* 12(6):e0179488.

Layla El Asri, Hannes Schulz, Shikhar Sharma, Jeremie Zumer, Justin Harris, Emery Fine, Rahul Mehrotra, and Kaheer Suleman. 2017. Frames: a corpus for adding memory to goal-oriented dialogue systems. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*. Association for Computational Linguistics, pages 207–219. http://aclweb.org/anthology/W17-5526.

Mohamed Elhoseiny, Babak Saleh, and Ahmed Elgammal. 2013. Write a classifier: Zero-shot learning using purely textual descriptions. In *Computer Vision (ICCV), 2013 IEEE International Conference on*. IEEE, pages 2584–2591.

Mihail Eric and Christopher D. Manning. 2017. Key-value retrieval networks for task-oriented dialogue. *CoRR* abs/1705.05414. http://arxiv.org/abs/1705.05414.

Shizhu He, Cao Liu, Kang Liu, and Jun Zhao. 2017. Generating natural answers by incorporating copying and retrieving mechanisms in sequence-to-sequence learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. volume 1, pages 199–208.

Joseph E. Hoag. 2008. *Synthetic Data Generation: Theory, Techniques and Applications*. Ph.D. thesis, Fayetteville, AR, USA. AAI3317844.

Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Synthetic data and artificial neural networks for natural scene text recognition. *CoRR* abs/1406.2227. http://arxiv.org/abs/1406.2227.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *CoRR* abs/1603.01360. http://arxiv.org/abs/1603.01360.

Mohammad Norouzi, Tomas Mikolov, Samy Bengio, Yoram Singer, Jonathon Shlens, Andrea Frome, Greg Corrado, and Jeffrey Dean. 2013. Zero-shot learning by convex combination of semantic embeddings. *CoRR* abs/1312.5650. http://arxiv.org/abs/1312.5650.

Emanuele Pianta, Christian Girardi, and Roberto Zanoli. 2008. The textpro tool suite. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*. Marrakech, Morocco. Http://www.lrec-conf.org/proceedings/lrec2008/.

Lance A. Ramshaw and Mitchell P. Marcus. 1995. Text chunking using transformation-based learning. *CoRR* cmp-lg/9505040. http://arxiv.org/abs/cmp-lg/9505040.

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

M. Schuster and K.K. Paliwal. 1997. Bidirectional recurrent neural networks. *Trans. Sig. Proc.* 45(11):2673–2681. https://doi.org/10.1109/78.650093.

Pararth Shah, Dilek Hakkani-Tür, Gökhan Tür, Abhinav Rastogi, Ankur Bapna, Neha Nayak, and Larry P. Heck. 2018. Building a conversational agent overnight with dialogue self-play. *CoRR* abs/1801.04871.

Richard Socher, Milind Ganjoo, Christopher D Manning, and Andrew Ng. 2013. Zero-shot learning through cross-modal transfer. In *Advances in Neural Information Processing Systems 26*, Curran Associates, Inc., pages 935–943. http://papers.nips.cc/paper/5027-zero-shot-learning-through-cross-modal-transfer.pdf.

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*. Association for Computational Linguistics, pages 142–147.

Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M Rush, Bart van Merriënboer, Armand Joulin, and Tomas Mikolov. 2015. Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698* .

Sihong Xie, Shaoxiong Wang, and Philip S. Yu. 2016. Active zero-shot learning. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, New York, NY, USA, CIKM '16, pages 1889–1892. https://doi.org/10.1145/2983323.2983866.

Appendix

In this section we provide some examples where NN_g is able to handle cases of entity names that other approaches are not able to. These cases are mainly due to token type ($type_1$ and $type_2$) and consecutive entities in a sentence – see table 7.

	NN_g	NN_p	Rule-based
$Type_1$ token error			
roasted	B-	B-	B-
asparagus	I-	I-	I-
with	I-	O	I-
orange	I-	B-	I-
glaze	I-	I-	I-
ann	B-	O	B-
chair	I-	B-	I-
mustard	I-	I-	I-
$Type_2$ token error			
dolly	B-	B-	B-
cushion	I-	I-	I-
cover	I-	O	I-
beige	I-	B-	I-
Consecutive entities error			
layene	B-	B-	B-
armchair	I-	I-	I-
bed	I-	I-	I-
brown	I-	I-	I-
trap	B-	I-	I-
chair	I-	I-	I-
dark	I-	I-	I-
brown	I-	I-	I-
ralf	B-	I-	I-
chair	I-	I-	I-
and	O	O	I-
malira	B-	B-	B-
table	I-	I-	I-

Table 7: some entity names correctly segmented by our approach but not by other approaches. In bold the $type_{1/2}$ token causing the error.