What's in a Food Name: Knowledge Induction from Gazetteers of Food Main Ingredient

Bernardo Magnini[1], **Vevake Balaraman**[1,2], **Simone Magnolini**[1,3], **Marco Guerini**[1]
[1] Fondazione Bruno Kessler, Via Sommarive 18, Povo, Trento — Italy
[2] University of Trento, Italy. [3] AdeptMind Scholar
{magnini, balaraman, magnolini, guerini}@fbk.eu

Abstract

English. We investigate head-noun identification in complex noun-compounds (e.g. *table* is the head-noun in *three legs table with white marble top*). The task is of high relevancy in several application scenarios, including utterance interpretation for dialogue systems, particularly in the context of e-commerce applications, where dozens of thousand of product descriptions for several domains and different languages have to be analyzed. We define guidelines for data annotation and propose a supervised neural model that is able to achieve 0.79 F1 on Italian food noun-compounds, which we consider an excellent result given both the minimal supervision required and the high linguistic complexity of the domain.

Italiano. *Affrontiamo il problema di identificare head-noun in nomi composti complessi (ad esempio "tavolo" is the head-noun in "tavolo con tre gambe e piano in marmo bianco"). Il compito é di alta rilevanza in numerosi contesti applicativi, inclusa l'interpretazione di enunciati nei sistemi di dialogo, in particolare nelle applicazioni di e-commerce, dove decine di migliaia di descrizioni di prodotti per vari domini e lingue differenti devono essere analizzate. Proponiamo un modello neurale supervisionato che riesce a raggiungere lo 0.79 di F-measure, che consideriamo un risultato eccellente data la minima quantitá di supervisione richiesta e la alta complessitá linguistica del dominio.*

1 Introduction

Noun-compounds are nominal descriptions that hold implicit semantic relations between their constituents (Shwartz and Dagan, 2018). For instance, an *apple cake* is a cake made of apples. While in the literature there has been a large interest in interpreting noun-compounds by classifying them with a fixed set of ontological relations (Nakov and Hearst, 2013), in this paper we focus on automatic recognition of the head-noun in noun-compounds. We assume that in each noun-compound there is a noun which can be considered as the more informative, as it brings the most relevant information that allows the correct interpretation of the whole noun-compound. For instance, in the *apple cake* example, we consider *cake* as the head-noun, because it brings more information than *apple* about the kind of food the compound describes (i.e. a dessert), its ingredients (i.e. likely, flour, milk and eggs), and the typical amount a person may eat (i.e. likely, a slice). While in simple noun-compounds the head-noun usually corresponds to the syntactic head of the compound, this is not the case for complex compounds, where the head-noun can occur in different positions of the compound, making its identification challenging. As an example, in the Italian food description *filetto di vitellone senza grasso visibile*, there are three nouns (i.e. *filetto*, *vitellone* and *grasso*) which are candidates to be the head-noun of the compound.

There are a number of tasks and application domains where identifying noun-compound head-nouns is relevant. A rather general context is ontology population (Buitelaar et al., 2005), where entity names automatically recognized in text are confronted against entity names already present in an ontology, and have to be appropriately matched in the ontology taxonomy. Our specific application interest is conversational agents for the e-commerce domain. Particularly, understanding names of products (e.g. food, furniture, clothes, digital equipment) as expressed by users in different languages, requires the capacity to distinguish

the main element in a product name (e.g. a *table* in *I am looking for a three legs table with white marble top*), in order to match them against vendor catalogues and to provide a meaningful dialogue with the user. The task is made much more challenging by the general lack of annotated data, so that fully supervised approaches are simply not feasible. Along this perspective, the long term goal of our work is to develop unsupervised techniques that can identify head-nouns in complex noun-compounds by learning properties on the base of the noun-compounds included in, possibly large, gazetteers, regardless of the domain and language in which they are described.

In this paper we propose a supervised approach based on a neural sequence-to-sequence model (Lample et al., 2016) augmented with noun-compound structural features (Guerini et al., 2018). This model identifies the more informative token(s) in the noun-compound, that are finally tagged as the head-noun. We run experiments on Italian food names, and show that, although the domain is very complex, results are promising.

The paper is structured as follow: we first define noun-compound head-noun identification, with specific reference to complex noun-compound (Section 2). Then we introduce the neural model we have implemented (Section 3), and finally the experimental setting and the results we have obtained (Section 4).

2 Food Compound-Nouns

In this Section we focus on Italian compound-nouns referring to food, the domain on which we run our experiments. Similar considerations and same methodology can be applied to compound-nouns in different domains and languages.

There is a very high variety of food compound-nouns, describing various aspects of food, including: simple food names, like *mortadella di fegato*, *pesce*, *gin and tonic*, *aglio fresco*; recipes mentioning their ingredients, like *scaloppine al limone*, *spaghetti al nero*, *passato di pollo*, *decotto di carciofo*; recipes focusing on preparation style, like *mandorle delle tre dame*, *cavolfiore alla napoletana*; food names focusing on visual or shape properties, like *filetto di vitellone senza grasso visibile*, *palline di formaggio fritte*; food descriptions containing a course name, like *antipasto di capesante*, *dessert di mascarpone*; food using fantasy names, like *frappé capriccioso*, or *insalata arlecchino*; food including proper names or brands, like *saint-honoré*, *tagliatelle Matilde*, *formaggio bel paese*; food names focusing on cooking modalities, like *pane fatto in casa*, or *peperoni fritti*; and focusing on alimentary properties, like *ragù di carne dietetico*, or *sangria analcolica*.

We assume that the head-noun of a food description is the more informative noun in the noun-compound, i.e. the noun that better allows to answer questions about properties of the food being described by the noun-compound. We consider the following four property related questions, in order of relevance:

1. What *food category* (e.g. meat, vegetable, cake, soup, pasta, fish, liquid, salad, etc.) is described by the noun-compound?
2. What *course* (e.g. main, appetizer, side dish, dessert, etc.) is described by the noun-compound?
3. Which is the *main ingredient* (in term of quantity) described by the noun-compound?
4. Which could be the overall *quantity* (expressed in grams) of food described by the noun-compound?

Although our approach does not require any domain knowledge, for the purpose of human annotation and evaluation it is useful to assume a simple ontology for food, where we define the properties used for judging head-nouns and the set of possible values for each property. Table 1 reports the food ontology at the base of our work.

Property	Values
Food category	meat, vegetable, cake, soup, pasta, fish, liquid, salad...
Course	main, first, second, appetizer, side , dessert...
Main ingredient	<simple food>
quantity	<grams>

Table 1: Food Ontology.

A good head-noun should be as much informative as possible about the noun-compound properties, or, in other terms, it should allow to infer as much as possible answers to questions 1-4. Answers to such questions are in most of the cases

graduated and probabilistic, as a noun-compound contains just a fraction of the knowledge needed to answer them. For instance, given question 1) for the food noun-compound *insalata noci e formaggio* should be posed in the following way: knowing that *formaggio* is part of a food description, which is the probability that the overall description correctly refers to a food of category *salad*? When the probability is very low, we assume a "no guess" value for the answer.

The core procedure for human annotations considers each content word in a food description, fills in the values of the four attributes, and then select the noun with the best guesses. Below some examples (in black the selected head of the food description):

- ***insalata*** *noci e formaggio*: because *insalata* is a better predictor of the food category than *formaggio* or *noci*.
- *involtini di* ***peperoni***: because *peperoni* is a better predictor of food category (i.e. vegetable) and of the main ingredient than *involtini*.
- ***budino*** *al cioccolato fondente*: because *budino* is a good predictor of food category (i.e. dessert) and a better predictor than *cioccolato* of the main ingredient (i.e. milk) of the noun-compound.

2.1 Task and Data Set

Given a food noun-compound, the task we address is to predict its head-noun, labelling one or more consecutive tokens in the food description. We assume that a head is always present, even in case it is poorly informative.

Two annotators were selected to annotate a data set of 436 food names, collected from recipe books, with their head-noun. The inter annotator agreement, computed at the token level, is Cohen's kappa: 0.91, which is considered very high.

In table 2 we give an overview of the data set of food-description head (FDH) we created focusing on two main orthogonal characteristics: whether the head-noun is comprised of a single token or of a multi-token, and whether the head-noun corresponds to the beginning of the food description or not. As can be seen, the vast majority of head-nouns is either made of a single token (almost 90% of cases), or starts at the beginning of the entity name (almost 80% of cases). The combination of the two accounts for roughly 70% of the cases. From the point of view of predicting the head-noun of a food name, easier cases are given by single token in first position, while harder cases are given by multi-token head inside the food name.

	FDH type		
Position	Single token	Multi token	Total
1^{st} token	72.48	9.17	81.65
Not 1^{st} token	17.89	0.46	18.35
Total	90.37	9.63	

Table 2: Coverage on the data set of head-noun characteristics (in %): either single token or multi-token and whether appearing at the beginning of the food description or not.

3 Model

The architecture we use to recognize head-nouns is based on a bidirectional LSTM (Long Short Term Memory) network (Graves and Schmidhuber, 2005), similar to the one presented in (Lample et al., 2016). We briefly describe the LSTM model used in the approach and proceed with the implementation details.

3.1 LSTM

Recurrent Neural Network (RNN) is a class of artificial neural network that resemble a chain of repeating modules to efficiently model sequential data (Mikolov et al., 2010). They take sequential data $(x_1, x_2, ...x_n)$ as input and provide a representation $(h_1, h_2, ...h_n)$ which captures the information at every time step in the input. Formally,

$$h_t = f(Ux_t + Wh_{t-1})$$

where x_t is the input at time t, U is the embedding matrix, f is a non-linear operation (such as sigmoid, tanh or ReLU) and W is the parameter of RNN learned during training.

The hidden state h_t of the network at time t captures only the left context of the sequence for the input at time t. The right context for the input at time t can be captured by performing the same operation in the negative time direction. The input can be represented by both its left context $\overrightarrow{h_t}$ and right context $\overleftarrow{h_t}$ as $h_t = [\overrightarrow{h_t}; \overleftarrow{h_t}]$. Similarly, the representation of the completed sentence is given by $h_T = [\overrightarrow{h_T}; \overleftarrow{h_0}]$. Such processing of the input in both forward and backward time-step is known as bidirectional RNN. Though a vanilla RNN is good

at modelling sequential data, it struggles to capture the long-term dependencies in the sequence. Long Short Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) is a special kind of RNN that is designed specifically to capture the long-term dependencies in sequential data. They compute the the hidden state h_t as follows,

$$
\begin{aligned}
i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i \\
f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f \\
\tilde{C}_t &= tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \\
C_t &= f_t * C_{(t-1)} + i_t * \tilde{C}_t \\
o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o \\
h_t &= o_t * tanh(C_t)
\end{aligned}
$$

where x_t is the embedding for input at time t; i_t, f_t, o_t are the input, forget and output gates, respectively.

3.2 Implementation

The task of head-noun identification aims to predict a sequence of tags $y = \{y_1, y_2, .., y_n\}$ given an input sequence $X = \{x_1, x_2, ..x_n\}$. The system is modeled as a sequence labelling task and consists of three main steps: i) *word embedding*: each word in the sequence is embedded to a higher dimension; ii) *Input encoder*: encoding the sequence of embeddings; iii) *Classification*: labelling the sequence.

Word embeddings. Each word in the input sequence is represented by a vector of d-dimensions that captures the syntactic and semantic information of the word. The representation is carried by a word embedding matrix $E \in \mathbb{R}^{d \times |v|}$ where $|v|$ is the input vocabulary size. In addition to this, the model combines a character embedding that is learned during training using a Bi-LSTM network to deal with out of vocabulary terms and possible misspellings (Ling et al., 2015).

To represent the core structure of a complex noun-compound, we also use the following handcrafted features of a head-noun candidate token (Guerini et al., 2018): (i) the actual position of the token within the compound name; (ii) the length of the candidate token; (iii) the frequency of the token in the gazetteer; (iv) the average length of the noun-compounds in the gazetteer containing the token; (v) the average position of the token in the noun-compound it appears in; (vi) the bigram probability with reference to the previous token in the noun-compound; (vii) if the token can be an noun-compound; (viii) the ratio of the time the token is the first token in a noun-compound; (ix) the ratio of the time the token is the last token in a noun-compound. These handcrafted features for each word are extracted from a large corpus of Italian food names reported in (Guerini et al., 2018).

The concatenation of word embedding, final states of bidirectional character embeddings network, and hand crafted features is used as the word representation.

Input encoder. LSTM nodes are used to encode the input sequence of word embeddings. We employ a bidirectional LSTM (Bi-LSTM) to capture the context in both forward and backward timesteps. The hidden representation of a word at time t is given as,

$$h_t = [\overrightarrow{h}_t; \overleftarrow{h}_t]$$

Classification. The output layer receives the hidden representation from the Bi-LSTM and outputs a probability distribution over the possible tag sequences. Then, a conditional random field (CRF) layer (Lafferty et al., 2001) is used to model the dependency in labelling tags. The hidden representations from the Bi-LSTM are passed through a linear layer to obtain the score P_i for each word in the input sequence $X = \{x_1, x_2, .., x_n\}$. The score for each possible output tag sequence $\hat{y} \in \hat{\mathbf{Y}}$ is then obtained as follows,

$$Score(\hat{y}) = \sum_{i=0}^{n} A_{y_i, y_{i+1}} + \sum_{i=1}^{n} P_{i, y_i}$$

where A is the transition matrix representing the transition scores from tag i to tag j. The probability of the tag sequence is then computed using a softmax operation as follows,

$$p(\hat{y}|X) = \frac{exp(Score(\hat{y}))}{\sum_{\tilde{y} \in \hat{\mathbf{Y}}} exp(Score(\tilde{y})}$$

The training is done by maximizing the log probability of the correct output tag sequence.

4 Experiments and Results

4.1 Setup

The dimension of character embedding is set to 30 and embeddings are learned using 50 hidden units

in each direction. For the word embeddings, as learning this level of representation with a small dataset is highly inefficient, we decided to use pre-trained embeddings trained using skip-gram (Mikolov et al., 2013) on the Italian corpus of Wikipedia. The input encoder consists of 120 hidden units in each direction with a dropout (E. Hinton et al., 2012) of 0.5 applied between the Bi-LSTM layer and the output layer.

4.2 Baselines

To compare the performance of the proposed approach, we provide two baselines: i) *1st token*, where the 1st token of a noun-compound is chosen as its head-noun; ii) *Spacy*[1], where the root token of the dependency tree for the noun-compound is chosen as its head-noun.

1st token. This baseline implicitly accounts for a number of linguistic behaviours of head-nouns in Italian language: (a) avoids stop words as head-nouns, as they do not occur at the first position of a noun-compound; (b) avoids adjectives as head-nouns, as they usually occur after the noun they modify; (c) captures the syntactic head of the noun-compound, which, in Italian is likely to be the first noun in a Noun Phrase; as already seen in Table 2. Summing up, the first-token baseline captures relevant linguistic behaviours, and is a strong competitor of our neural model, as in more than 80% of the entries in our dataset the first token belongs to head-noun of the noun-compound.

Spacy. This is a widely known open-source library for natural language processing and include a syntactic dependency parser. Given an input sequence, based on the result returned by the dependency parser, the root of the sequence is chosen to be the head-noun. We used the statistical model *it_core_news_sm*[2] released by Spacy for Italian language.

4.3 Evaluation metric

The performance of the models are evaluated using F1 score as in CoNLL-2003 NER evaluation (Sang and Meulder, 2003), which is a standard for evaluating sequence tagging tasks.

4.4 Results

The results for the FDH dataset are shown in Table 3. The baselines *1st token* and *Spacy* achieve a performance of 70.27 of 62.67 respectively. In particular, the performance of syntactic dependency parser from Spacy reiterates the difference between the semantic and syntactic head. The results are shown by incremental features, for the proposed approach. The models reported without CRF, are trained using a softmax function as output layer to predict the tag. We can notice from the results that using only the pre-trained embeddings, the network suffers from a poor recall and fails to achieve even the baseline performance. However, using either character embedding or the hand-crafted features, improves the performance of the model on par with the baseline. Since the single token head-noun in FDH dataset is very high (as shown in table 2), learning the multi token head-nouns and the dependency of tags is a challenge. However, introducing the CRF layer to jointly predict the sequence of tags in combination with the hand crafted features, enables us to predict multi-token heads and improve the performance of the model to 78.09. Finally, the character embeddings learned during training helps to improves the recall further, reaching a F1 score of 79.58.

	Accuracy	Precision	Recall	F1
Baselines				
1st token	83.74	70.29	70.24	70.27
Spacy	78.47	62.70	62.67	62.67
Bi-LSTM				
a) word_emb	84.06	74.10	65.18	69.28
b) a + hc_feat	85.17	75.76	66.50	70.76
c) a + char_emb	85.21	76.24	66.28	70.79
d) b + CRF	88.07	78.57	77.67	78.09
d) d + char_emb	**88.59**	**80.58**	**78.62**	**79.58**

Table 3: Experimental results on FDH dataset.

5 Conclusion and Future Work

We have addressed head-noun identification in complex noun-compounds, a task of high relevancy in utterance interpretation for dialogue systems. We proposed a neural model, and experiments on Italian food noun-compounds show that the model is able to outperform strong baselines even with a small amount of data. For the future we plan to extend our investigation to other domain and languages.

References

Paul Buitelaar, Philipp Cimiano, and Bernardo Magnini. 2005. *Ontology Learning from Text:*

[1] https://spacy.io/

[2] https://spacy.io/models/it

Methods, Evaluation and Applications, volume 123 of *Frontiers in Artificial Intelligence and Applications Series*. IOS Press, Amsterdam, 7.

Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R. Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. arXiv, 07.

A. Graves and J. Schmidhuber. 2005. Framewise phoneme classification with bidirectional lstm networks. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 4, pages 2047–2052 vol. 4, July.

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, Melbourne, Australia, July.

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

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

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

Wang Ling, Chris Dyer, Alan W. Black, Isabel Trancoso, Ramon Fermandez, Silvio Amir, Luís Marujo, and Tiago Luís. 2015. Finding function in form: Compositional character models for open vocabulary word representation. In *EMNLP*.

Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černocký, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Proceedings of the 11th Annual Conference of the International Speech Communication Association (INTERSPEECH 2010)*, volume 2010, pages 1045–1048. International Speech Communication Association.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Preslav Nakov and Marti A. Hearst. 2013. Semantic interpretation of noun compounds using verbal and other paraphrases. *TSLP*, 10(3):13:1–13:51.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *CoRR*, cs.CL/0306050.

Vered Shwartz and Ido Dagan. 2018. Paraphrase to explicate: Revealing implicit noun-compound relations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1200–1211. Association for Computational Linguistics.