Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 2110–2119
Marseille, 11–16 May 2020

Comparing Machine Learning and Deep Learning Approaches on NLP Tasks for the Italian Language

Bernardo Magnini, Alberto Lavelli, Simone Magnolini
Fondazione Bruno Kessler
via Sommarive 18, Povo - Trento (ITALY)
{magnini, lavelli, magnolini}@fbk.eu

Abstract

We present a comparison between deep learning and traditional machine learning methods for various NLP tasks in Italian. We carried on experiments using available datasets (e.g., from the Evalita shared tasks) on two sequence tagging tasks (i.e., named entity recognition and nominal entity recognition) and four classification tasks (i.e., lexical relations among words, semantic relations among sentences, sentiment analysis and text classification). We show that deep learning approaches outperform traditional machine learning algorithms in sequence tagging, while for classification tasks that heavily rely on semantics approaches based on feature engineering are still competitive. We think that a similar analysis could be carried out for other languages to provide an assessment of machine learning / deep learning models across different languages.

Keywords: Machine Learning, Deep Learning, Italian Language

1. Introduction

In the recent years, the so called "deep learning revolution" has influenced and changed many fields of Artificial Intelligence (e.g., machine learning and computer vision) and has also affected all areas related to human language technologies. Initial results have been obtained with the adoption of deep neural networks in speech recognition, with a significant boost of performance in automatic speech recognition systems (Graves et al., 2013). In Machine Translation, starting from 2013, the phrase-based statistical approaches that were at the state of the art have been gradually substituted with neural machine translation, based on deep learning architectures, which have been proven to obtain better performance (Bahdanau et al., 2014). The main reason for this increase in performance is that, as more training data are available both for speech recognition and machine translation, large neural networks have been proven to be superior to traditional machine learning (ML) algorithms, such as support vector machines.

However, if we consider tasks related to semantic analysis of text, the limited availability of semantically annotated data, typically requiring specialized human effort, has slowed the adoption of neural approaches. It is only in the last few years that deep learning has obtained high performance across different NLP tasks. These models can often be trained with a single end-to-end model and do not require task-specific feature engineering, thus they not only tend to perform better than traditional ML, but they do require less human effort, making their adoption convenient.

In this paper we provide a

comparison between traditional approaches and deep learning applied to NLP tasks in the area of information extraction from Italian texts. We carried on experiments using available datasets on both sequence tagging (i.e., named entity recognition, nominal entity recognition) and classification tasks (i.e., lexical relations among words, semantic relations among sentences, sentiment analysis, text classification).

We consider this paper as a contribution in the direction of developing benchmarks encompassing a variety of tasks in order to favour models that share general linguistic knowledge across tasks. This is very much in the spirit of GLUE, the General Language Understanding Evaluation (Wang et al., 2018), a collection of resources for training, evaluating, and analyzing natural language understanding systems.

The paper is structured as follows. Section 2 reports basic notions about deep learning for NLP that will be used for our experiments. Sections 3 and 4 focus on sequence tagging tasks, named entity recognition and nominal entity recognition, respectively. Sections 4-7 report on classification tasks: lexical relations, textual entailment, sentiment analysis and text classification. Finally, Section 9 discusses our achievement and proposes work for the future.

2. Deep Learning for NLP

This section provides basic notions on deep learning for NLP, which will be used in the rest of the paper. We introduce word vector representations, pre-trained language models, and long-short-term-memory architectures.

2.1. Word Embeddings

Word embeddings are essentially vector representations of words, that are typically learnt by an unsupervised model when fed with large amounts of text (e.g., Wikipedia, scientific literature, news articles, etc.). These representations capture semantic similarity between words among other properties. They are hence very useful to represent words in downstream NLP tasks such as POS tagging, NER etc.
Three families of word embeddings can be identified:

- Bag of words based. The original word order independent models like Word2vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014).
- Attention (Transformer) based. Embeddings generated by BERT (Devlin et al., 2018), which has produced state-of-the-art results to date in downstream

tasks like NER, Q&A, classification etc. BERT takes into account the order of words in a sentence but is based on attention mechanism as opposed to sequence models like ELMo.

- RNN family based. Sequence models (ELMo) that produce word embeddings (Peters et al., 2018). ELMo uses stacked bidirectional LSTMs to generate word embeddings that have different properties based on the layer that generates them.

2.2. BERT

BERT (Devlin et al., 2018) is a deep learning model that has given state-of-the-art results on a wide variety of natural language processing tasks. It stands for Bidirectional Encoder Representations for Transformers. It has been pre-trained on Wikipedia and BooksCorpus and requires task-specific fine-tuning.
BERT is available pre-trained on domain-specific corpora. E.g., Clinical BERT (BERT pre-trained on a corpus of clinical notes) and sciBERT (Pre-Trained Contextualized Embeddings for Scientific Text). BioBERT (Lee et al., 2019) (Bidirectional Encoder Representations from Transformers for Biomedical Text Mining) is a domain-specific language representation model pre-trained on large-scale biomedical corpora. With almost the same architecture across tasks, BioBERT largely outperforms BERT and previous state-of-the-art models in a variety of biomedical text mining tasks when pre-trained on biomedical corpora. While BERT obtains performance comparable to that of previous state-of-the-art models, BioBERT significantly outperforms them on the following three representative biomedical text mining tasks: biomedical named entity recognition (0.62% F_1 score improvement), biomedical relation extraction (2.80% F_1 score improvement) and biomedical question answering (12.24% MRR improvement).
BERT is also available for languages other than English[1]. In particular, it is provided a model for Chinese and a single model for all the other languages, including Italian.

2.3. A Sequence Labeling Neural Architecture: *NeuroNLP2*

In this section we introduce *NeuroNLP2* (Ma and Hovy, 2016), a reference neural architecture for sequence labeling in NLP that achieved state-of-the-art performance for named entity recognition for English on the ConLL-2003 dataset. Specifically, we describe the most recent implementation of the system in Pytorch distributed by the authors[2]. We selected this system not only for its state-of-the-art performance and for code availability, but also for the peculiar structure of the network, which is common to other works, including (Lample et al., 2016). 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 LSTMs; (iii) a CRF layer that decodes the best label sequence.

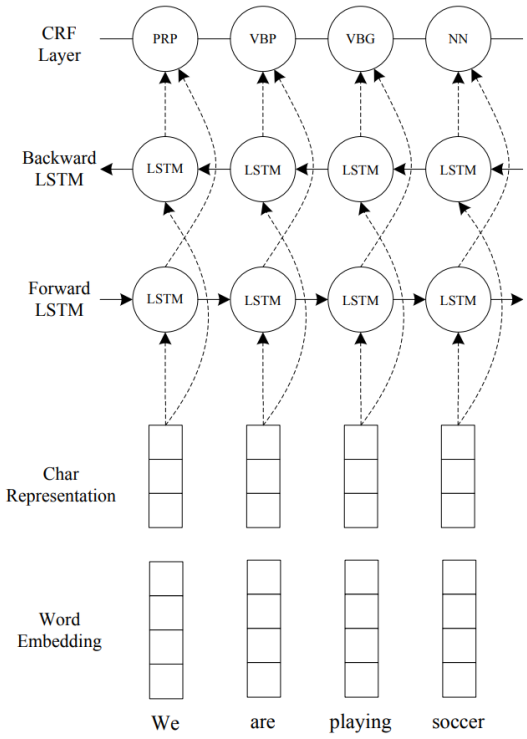

Figure 1: The main NeuroNLP2 structure. Dashed arrows indicate dropout layers applied on both the input and output vectors of BLSTM.

NeuroNLP2 constructs a neural network model by feeding the output vectors of BLSTM into a CRF layer, as it is depicted in Figure 1. 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, 2016), except that it uses only character embeddings as inputs, without character type.
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. A LSTM unit is composed of three multiplicative gates which control the proportions of information to forget and to pass on to the next time step. The basic idea is to present each sequence forwards and backwards to two separate LSTMs and then to concatenate the output to capture past and future information, respectively.
The LSTM's 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 fu-

[1] https://github.com/google-research/bert/blob/master/multilingual.md
[2] https://github.com/XuezheMax/NeuroNLP2

ture (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 of the BLSTM.
Finally, the third layer implemented in 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.

3. Named Entity Recognition

Named entities are proper names referring to persons, locations and organizations. A reference paper for the application of deep learning techniques to Named Entity Recognition is Ma and Hovy (2016), whose approach, NeuroNLP2, has been presented in Section 2.3.. The system is truly end-to-end, requiring no feature engineering or data pre-processing, thus making it applicable to a wide range of sequence labeling tasks. They evaluate the system on two datasets for two sequence labeling tasks — PennTreebank WSJ corpus for part-of-speech tagging and the CoNLL 2003 corpus for named entity recognition (NER). They obtain state-of-the-art performance on both datasets — 97.55% accuracy for part-of-speech tagging and 91.21% F_1 for NER.
Neural architectures for Italian NER have been already investigated by several works. Bonadiman et al. (2015) introduce a Deep Neural Network (DNN) for Named Entity Recognizers (NERs) in Italian. The network uses a sliding window of word contexts to predict tags. It relies on a simple word-level log-likelihood as a cost function and uses a new recurrent feedback mechanism to ensure that the dependencies between the output tags are properly modeled. The evaluation on the Evalita 2009 benchmark (Speranza, 2009) shows that the DNN performs on par with the best NERs, outperforming the state of the art when gazetteer features are used.
Basile et al. (2017) propose a Deep Learning architecture for sequence labeling based on a state-of-the-art model that exploits both word- and character-level representations through the combination of bidirectional LSTM, CNN and CRF. They evaluate the proposed method on three NLP tasks for Italian: PoS-tagging of tweets, Named Entity Recognition and Super-Sense Tagging. Results show that the system is able to achieve state-of-the-art performance in all the tasks and in some cases overcomes the best systems previously developed for Italian.
Magnolini et al. (2019) 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 a gazetteer and then concatenating such features with the input embeddings of a neural model is the best strategy in all experimental settings, significantly outperforming more conventional approaches. For the experiments they used exactly the same network parameters described in Ma and Hovy (2016) and provided as default by the available implementation. As input embeddings they 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 they use Stanford's GloVe 50-dimensional embeddings trained on a Wikipedia's dump [3] with the default setup. For out-of-vocabulary words they use a unique randomly generated vector for every word.

dataset	BERT	Best Evalita 2009	SotA
NER Evalita 2009	**85.05**	82.00	84.33

Table 1: Application of BERT fine-tuning for Named Entity Recognition.

In Table 1 we report the results we obtained applying BERT (multilingual model) to the NER Evalita 2009 (Speranza, 2009) task. The BERT model is compared against the system that obtained the best result at Evalita 2009, (Zanoli et al., 2009) and the state of the art for Italian NER (Nguyen et al., 2010).
As suggested by BERT developers, for sequence labeling BERT-NER[4] was used simply performing some fine tuning on the training data with default parameters. However, in order to obtain this result the default parameters are different from the ones used for classification. Another important detail is the great difference in performance among the case sensitive and the case insensitive model: the former one outperforms in a significant way the latter one.
It can be noticed that the purpose of the experiment is not to obtain a new state of the art (even if in this case it was achieved), but to investigate how deep learning performs on a different task in a language that is not English. In fact, BERT-NER is not the implementation presented in Devlin et al. (2018), but a third party implementation slightly less performing than the one presented in the paper, that is not freely available.

4. Nominal Entity Recognition

Nominal entities are noun phrase expressions describing an entity. They can be composed by a single noun (e.g., *pasta*, *carpet*, *parka*) or by more than one token (e.g., *capri sofa bed beige*, *red jeans skinny fit*, *light weigh full frame camera*, *grilled pork belly tacos*). Differently from named entities, 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*).

[3] 20/04/2018
[4] `https://github.com/kyzhouhzau/BERT-NER`

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 2: Example of IOB annotation of food nominal entities.

Nominal entity recognition has been approached with systems based on linguistic knowledge, including morpho-syntactic information, chunking, and head identification (Pianta and Tonelli, 2010). In the framework of the ACE program (Doddington et al., 2004) there has been several attempts to develop supervised systems for nominal entities (Haghighi and Klein, 2010), which, however, had to face the problem of the scarcity of annotated data, and, for this reason, were developed for few entity types.

Similarly to what is done for named entities, nominal 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 2. It is worth to mention that the IOB format does not allow to represent nested entities, a potential limitation for nominal entities.

4.1. Datasets for Nominal Entity Recognition

We use DPD – Diabetic Patients Diary – a dataset 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 DPD 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 3 we describe the gazetteers that we have used in our experiments for two datasets (DPD for nominal entities and CoNNL for named entities), 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 entity 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).

4.2. Experiments on Nominal Entity Recognition

In our experiments we compare nominal entity recognition on the DPD dataset against named entity recognition on the CoNNL dataset. In both cases we show four configurations: (i) NeuroNLP2, the neural architecture presented in Section 2.3; (ii) NeuroNLP2 with the use of gazetteer features (single-token) as reported in Table 3; (iii) NeuroNLP2 with the use of gazetteer features (multi-token); (iv) NeuroNLP2 with the use of gazetteer features based on a dedicated neural model (NN$_g$).

Table 4 shows the results of gazetteer integration as embedding. 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 F_1). The combination of NeuroNLP2 and NN$_g$ reaches state-of-the-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. 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.

5. Lexical Relations among Words

This section addresses the capacity of neural models to detect semantic relations (e.g., synonymy, semantic similarity, entailment, compatibility) between words (or phrases, like the nominal expressions described in Section 4.2). We focus our experiments on the *compatibility relation*, and adopt the definition of compatibility proposed by Kruszewski and Baroni (2015): two linguistic expressions w_1 and w_2 are compatible iff, in a reasonably normal state of affairs, they can both truthfully refer to the same thing. If they cannot, then they are incompatible. Under this definition compatibility is a symmetric relation, which is different both from subsumption, which in not symmetric, from semantic similarity (Agirre et al., 2012) (two expressions can be compatible although not semantically similar, like *aperitif* and *chips*, and from textual entailment (Dagan et al., 2005), as entailment is not a symmetric relation.

5.1. Task definition

The task is defined as follows: given a lexicon L and a query q, the system should retrieve and order all the terms l_i in L such that q and l_i are compatible. L is a finite set of n terms, and both terms l_i and the query q are nominal expressions composed of one or more words. Accordingly,

dataset	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 3: Gazetteers used in the experiments for Nominal Entity Recognition. 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	F_1	Accuracy	Precision	Recall	F_1
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 4: Results on Nominal Entity Recognition using gazetteers as features together with embeddings.

the expected output is a (possibly empty) list of compatible terms ordered by relevance with respect to q.

In practical scenarios (e.g., ontology matching) the lexicon L can be composed of thousands, or tens of thousand of terms (e.g., all concept names in DBPedia, all the names of products in a catalogue, all word forms in WordNet, or all the entry names in a dictionary). In our definition we do not consider any relation among terms (e.g., semantic relations such as IS-A), so that terms can be considered as independent. Finally, the problem is treated as in information retrieval (IR), assuming that queries are nominal expressions (as they are in most cases in IR) and that the document collection (i.e., our lexicon L) is composed of documents consisting of a single term (i.e., a nominal expression). Compatibility is formulated as a binary classification problem, as two expressions can be either compatible or incompatible. While Kruszewski and Baroni (2015) use a continuous scale from 1 (low compatibility) to 7 (high compatibility) and then estimate a compatibility threshold, in our work we use a three-value scale (from 1 to 3).

5.2. Datasets on compatibility relation

We focused our experiments on compatibility relations among food names. We adopted an existing ontology in the food domain, the HeLiS ontology (Bailoni et al., 2016)[5], which we use as the lexicon L. As for the queries q, we built a set of 100 query terms that are completely independent of those contained in HeLiS; in fact, we extracted them from among the dishes or types of food annotated in the Diabetic Patients Diary[6], a corpus of above 1,000 meal descriptions written by diabetic patients (for example, *wholemeal pasta with raw ham and tomatoes, cucumbers*).

For each query, the annotator was presented with a list of 5 to 10 terms in alphabetical order. The annotator had to annotate each term for compatibility with the query it was associated with, which means they had to decide whether the two given expressions could or could not refer to the same dish or food. More specifically, the task consisted of assigning to each term a compatibility rating on a 3-point scale where 3 means that they were fully convinced that the two expressions could refer to the same dish or food, while 1 means that they thought that it was impossible that the two expressions referred to the same dish or food. In the case of *chicken with mushrooms and onions* and *chicken with mushrooms*, for example, the expected compatibility rating is 3, since the two expressions can (easily) refer to the same dish (when mentioning a dish, people can easily omit secondary ingredients). On the other hand, annotators would assign a compatibility rating of 1 to the *cod fillet* and *pork fillet* pair, since a cod fillet cannot be a pork fillet. Finally, a compatibility rating of 2 would be assigned to pairs like *cod fillet with asparagus* and *rice cod fillet with fennel and capers*; in this case, the secondary ingredients listed in the query and in the proposed term differ, but they both refer to cod fillet. Inter-annotator agreement, computed in terms of kappa statistic on the dual annotation of a subset of 21 query terms (for a total of 184 terms), is 0.76.

For our experiments, we split the dataset in two parts; half of the data was used as a development set and half as a test set (see Table 5 for detailed statistics about the two datasets

	Dev	Test	Total
Total queries	50	50	100
Tokens/query	2.58	2.76	2.67
1 terms	223 (54.9%)	261 (60.7%)	484 (57.9%)
2 terms	156 (38.4%)	135 (31.4%)	291 (34.8%)
3 terms	27 (6.7%)	34 (7.9%)	61 (7.3%)
Total terms	406	430	836
Tokens/term	4.01	4.21	4.12
Terms/query	8.12	8.6	8.36

Table 5: Statistics about the dataset for compatibility relation (n $terms$ indicates the terms with compatibility rating equal to n).

[5] http://w3id.org/helis.

[6] https://hlt-nlp.fbk.eu/technologies/dpd.

and their annotations).

5.3. Experiments on compatibility relation

We conducted experiments with the algorithms below.

Semantic Similarity. A baseline based on similarity of word embeddings. A term is considered compatible with the query if it is ranked in the best 5 terms according to the cosine similarity of the vector representing the query and the vector representing the term. More precisely, we first extract the vector of each token of the query from a GloVe model (Pennington et al., 2014) and then compute the average of the extracted vectors (i.e., the centroid vector). This method partially includes token overlap, in fact equal tokens have equal vectors, so expressions composed of the same tokens have the same vector representation. This baseline exclusively uses the information carried by GloVe vectors, as all the vectors included in the model are used with the same weight.

Semantic Similarity with Threshold. The same as the semantic similarity baseline, with the addition of a compatibility threshold operating over the best 5 terms. A term is considered compatible with the query if it is ranked above the compatibility threshold, which is empirically is calculated on WordNet data.

Semantic Head. An approach based on the automatic recognition of the *semantic head* of a term, without any threshold. A term is considered compatible with the query if it is ranked in the best 5 terms according to both their respective semantic heads and the similarity of their tokens.

Semantic Head with Threshold. The approach based on semantic heads, integrated with a compatibility threshold over the best 5 terms retrieved by the semantic head algorithm.

5.4. Evaluation Metrics

Evaluation is based on Mean Reciprocal Rank (MRR) (Craswell, 2009), a standard measure to evaluate retrieval systems (particularly question answering). While MRR is designed for binary classification of retrieved objects (i.e., correct vs incorrect), in our scenario retrieved terms can assume one value on a 3-point compatibility scale. We therefore calculate the MRR for each value, thus obtaining MRR_1, MRR_2 and MRR_3, respectively the MRR of terms that are not compatible with query (value=1), of terms with low compatibility (value=2) and of terms that are fully compatible (value=3). Results in Table 6 are presented using the three metrics $MRR_{1,3}$, $MRR_{2,3}$ and MRR_1, as described below.

$MRR_{1,3}$ is the difference between MRR_3 and MRR_1, and indicates the ability of the system to rank compatible terms higher than the incompatible ones. This metric is the weighted average of the three MRR, with the MRR_2 weight set to 0 (i.e., [-1 0 1]). $MRR_{1,3}$ is normalized over [-1 1].

$MRR_{2,3}$ is the weighted average of the three MRR with the following respective weights [0 0.5 1], and it is meant to captures the ability of the algorithm to retrieve only the terms that compatible (value=3) or almost compatible (value=2). $MRR_{2,3}$ is normalized over [0 1].

MRR_1 (i.e., [1 0 0]) is meant to capture the capacity of the system to rank incompatible terms lower than all other terms. Although this information is also captured by $MRR_{1,3}$, MRR_1 alone highlights the effect that different algorithms have on the reduction of misclassifications. MRR_1 is normalized over [0 1].

5.5. Results on compatibility relation

Results (see Table 6) show that the semantic head approach systematically outperforms the semantic similarity baseline, both when the threshold is used and when it is not used. The algorithms with the threshold strongly reduce MRR_1, with a beneficial effect also on $MRR_{1,3}$; this is actually an expected effect of the threshold, as it enables the system to better distinguish between compatible and incompatible terms. On the other hand, a drawback of introducing the threshold is that it reduces $MRR_{2,3}$, i.e., the capability of the system to retrieve related terms. It is also interesting to notice that the decrease in $MRR_{2,3}$ is greater in the semantic similarity baseline, which is due to the fact that it implements only the relatedness threshold.

As a final consideration, we point out that the decrease in performance on the test set as compared to the development set is consistent for the semantic head approach in terms of all the metrics; this shows that the compatibility threshold is not overfitted on the development data, but it is general and has the same effect on the development and the test data.

Finally, in the last line of Table 6 we report the results obtained applying the multilingual model of BERT to the compatibility task. BERT was applied through fine-tuning of the multilingual model over the data of the compatibility task. In addition, some fine tuning for the task was performed on the generic model using the parameters suggested in Devlin et al. (2018). BERT performs better than the previous approaches based on semantic similarity among vectors, confirming the high capacity of the BERT model to capture semantic relations among words, even for Italian.

6. Textual Entailment

Driven by the assumption that language understanding crucially depends on the ability to recognize semantic relations among portions of text, several text-to-text inference tasks have been proposed in the last decade, including recognizing paraphrasing (Dolan and Brockett., 2005), recognizing textual entailment (RTE) (Dagan et al., 2005), and semantic similarity (Agirre et al., 2012). A common characteristic of such tasks is that the input are two portions of text, let's call them $Text1$ and $Text2$, and the output is a semantic relation between the two texts, possibly with a degree of confidence of the system. For instance, given the following text fragments:

Example 1. Text1: *George Clooney's longest relationship ever might have been with a pig. The actor owned Max, a 300-pound pig.*
Text2: *Max is an animal.*

a system should be able to recognize that there is an "entailment" relation among $Text1$ and $Text2$.

	Development			Test		
Metric	$MRR_{1,3}$	$MRR_{2,3}$	MRR_1	$MRR_{1,3}$	$MRR_{2,3}$	MRR_1
Weights	[-1 0 1]	[0 0.5 1]	[1 0 0]	[-1 0 1]	[0 0.5 1]	[1 0 0]
Semantic similarity	-0.236	0.345	0.398	-0.213	0.346	0.407
Semantic similarity with Threshold	-0.011	0.259	0.174	-0.028	0.231	0.179
Semantic head	-0.114	**0.396**	0.274	-0.184	**0.356**	0.357
Semantic head with Threshold	**0.034**	0.357	**0.111**	**-0.015**	0.298	**0.165**
BERT	-	-	-	-0.018	0.318	0.132

Table 6: Results on the development and test sets for compatibility relation detection.

6.1. Datasets used for Textual Entailment

We have tested the performance of a neural approach, based on BERT, on two RTE datasets available for Italian.

RTE3 Italian. This is the Italian translation of the RTE-3 dataset carried out during the EU project EXCITEMENT [7]. The RTE-3 dataset for English (Giampiccolo et al., 2007) consists of 1600 text-hypothesis pairs, equally divided into a development set and a test set. While the length of the hypotheses (h) was the same as in the RTE1a and RTE2 datasets, a certain number of texts (t) were longer than in previous datasets, up to a paragraph. Four applications – namely IE, IR, QA and SUM – were considered as settings or contexts for the pairs generation, and 200 pairs were selected for each application in each dataset.

RTE Evalita 2009. This is the dataset developed for Evalita 2009 (Bos et al., 2009) tasks. Pairs of texts have be taken from Italian Wikipedia articles, and are constructed by manually annotating contrasting texts taken from the version history as provided by Wikipedia. The following is a pair where Text1 entails Text2:

Example 2. Text1: *Parla di attivita' nei panni di direttore commerciale e, dopo sei mesi, di direttore generale.*
Text2: *Parla di attivita' di direttore commerciale e, dopo sei mesi, di direttore generale*

6.2. Results for textual entailment

In Table 7 we report the results obtained applying a neural model, BERT (multilingual), over the two datasets.

BERT Multilingual. This approach makes use of the BERT multilingual language model (Devlin et al., 2018) in order to establish as many as possible relations between Text1 and Text2. A threshold is then estimated on the training data, and used to separate entailment and no-entailment on the test data. As suggested by the BERT developers for classification tasks, some fine tuning for the task was performed on the generic model using the parameters suggested in Devlin et al. (2018).

State of the Art: EDITS. The system used for the experiments is the EDITS package (Edit Distance Textual Entailment Suite) (Kouylekov and Magnini, 2005) . EDITS implements a distance-based approach for recognizing textual entailment, which assumes that the distance between Text1 and Text2 is a characteristic that separates the positive sentence pairs, for which the entailment relation holds, from the negative pairs, for which the entailment relation does not hold. More specifically, EDITS is based on edit distance algorithms, and computes the T-H distance as the overall cost of the edit operations (i.e., insertion, deletion and substitution) that are necessary to transform Text1 into Text2.
In this case we have a mixed situation. In fact, while BERT achieves a significant improvement over the state of the art (i.e., the EDITS system - (Kouylekov and Magnini, 2005)), it is largely below the state of the art in the Evalita 2009 dataset. This is probably due to the fact that the variations between Text1 and Text2 in the Evalita dataset are only partially due to semantic phenomena, and as a consequence they are not captured by the BERT language model. On the other hand, the RTE3 dataset contains much more semantic lexical relations between the two sentences, and the BERT model seems to better capture such relation with respect to EDITS (+3.5), which is based on word relations in WordNet.

dataset	BERT (multilingual)	SotA
RTE 3 (ita)	**69.25**	63.50
RTE Evalita 2009	55.00	**71.00**

Table 7: Application of BERT to Textual Entailment.

7. Sentiment Analysis

Three shared tasks on sentiment analysis from Italian tweets were organized in the context of the EVALITA evaluation campaigns. SENTIPOLC (SENTIment POLarity Classification) was organized at EVALITA 2014 & 2016 (Basile et al., 2014; Barbieri et al., 2016). In 2016 the focus was on Italian texts from Twitter and there was a set of related tasks with an increasing level of complexity. The main task concerns sentiment polarity classification at the message-level. Sentiments expressed in tweets are typically categorized as positive, negative or neutral, but a message can contain parts expressing both positive and negative sentiment (mixed sentiment), a feature that should be tackled. ABSITA (Aspect-Based Sentiment analysis at EVALITA) was organized at EVALITA 2018 (Basile et al., 2018), as an evolution of Sentiment Analysis aiming at capturing the aspect-level opinions expressed in natural language texts. Aspect-based Sentiment Analysis is approached as a sequence of two subtasks: Aspect Category Detection (ACD) and Aspect Category Polarity (ACP).

[7] https://sites.google.com/site/excitementproject/results/RTE3-ITA_V1_2012-10-04.zip

In Table 8 we report the results obtained applying BERT (multilingual) to SENTIPOLC 2016 (task 2) and to ABSITA 2018 (tasks ACD and ACP). As suggested for classification tasks, some fine tuning for the task was performed on the generic model using the parameters suggested in Devlin et al. (2018). In particular, some aspects of the SENTIPOLC 2016 dataset are difficult to address with BERT. For example, the fact that the dataset is strongly unbalanced, usually an important aspect to take into account with a supervised system like BERT. To reduce this effect we down-sample the most common polarity, but even in this case, the result is not competitive with the state of the art.
On the other hand, is important to notice that in both cases (SENTIPOLC 2016 and ABSITA 2018), the models were not fine-tuned on Italian, but only on the task. According to the paper by Pires et al. (2019), multilingual BERT is able to perform some cross-lingual adaptation but it is reasonable to think that in a task more related to semantic a deeper process of fine-tuning is needed.

dataset	BERT	SotA
SENTIPOLC 2016 - Task 2	52.17	**66.38**
ABSITA 2018 - Task ACD	74.05	**81.08**
ABSITA 2018 - Task ACP	68.13	**76.73**

Table 8: Application of BERT to Sentiment Analysis.

8. Text Classification

Finally, we focus on text classification applied to radiological reports in Italian. Radiological reporting generates a large amount of free-text clinical narratives, a potentially valuable source of information for improving clinical care and supporting research. The use of automatic techniques to analyze such reports is necessary to make their content effectively available to radiologists in an aggregated form. In (Gerevini et al., 2018) the focus is on the classification of chest computed tomography reports according to a classification schema proposed for this task by radiologists of the Italian hospital ASST Spedali Civili di Brescia. The system is built exploiting a training dataset containing reports annotated by radiologists. Each report is classified according to the schema developed by radiologists and textual evidences are marked in the report. The annotations are then used to train different machine learning based classifiers. A method based on a cascade of classifiers which make use of a set of syntactic and semantic features is presented. The resulting system is a novel hierarchical classification system for the given task, that was experimentally evaluated.
As a follow-up of the work reported in (Gerevini et al., 2018), in (Putelli et al., submitted) deep learning techniques and in particular Long Short Term Memory (LSTM) networks (currently, the state-of-the-art method for many Natural Language Processing tasks) are applied to the same task, without the use of textual annotations. Each report is classified using a combination of neural network classifiers which make use of syntactic and semantic features. The resulting system is a novel hierarchical classification system for the given task. In Table 9, there is a comparison with the performance of the system based on standard machine learning techniques and annotations of relevant snippets.

	Annotation		Deep Learning	
	Acc	FM	Acc	FM
Exam type	96.0	95.8	**96.2**	**96.0**
Result (First Exam)	77.3	76.1	**78.3**	**76.3**
Result (Follow-Up)	73.9	65.6	**81.9**	**71.9**
Lesion Nature	66.3	62.3	**73.2**	**71.2**
Site Lung	**93.2**	71.9	90.9	**76.6**
Site Pleura	93.2	75.5	**94.4**	**75.8**
Site Mediastinum	**92.9**	**81.0**	88.3	72.9

Table 9: Classification of radiological reports. Comparison between the approach based on standard ML techniques and textual annotations and the model based on deep learning. In boldface the best results.

9. Discussion and conclusions

We have presented a comparison between deep learning and traditional machine learning methods for various NLP tasks in Italian. We carried on experiments using available datasets on two sequence tagging tasks (i.e., named entity recognition and nominal entity recognition) and four classification tasks (i.e., lexical relations among words, semantic relations among sentences, sentiment analysis and text classification). Our experiments show that deep learning approaches outperform traditional machine learning algorithms in sequence tagging, while for classification tasks that heavily rely on semantics approaches based on feature engineering are still competitive. More in detail:

- BERT outperforms previous approaches both for named entities, textual entailment (RTE dataset) and text classification on clinical reports;
- on nominal entity recognition, a task much more complex than NER, we have shown that the NeuroNLP2 model can be extended with terms contained in a gazetteer, achieving state-of-the-art performance;
- on the three datasets for sentiment analysis on tweets traditional machine learning outperforms BERT, indicating that more accurate fine tuning is still necessary;
- on lexical relations (i.e., compatibility among words) a simple BERT fine tuning achieves results comparable to those obtained by more complex architectures using linguistic features (e.g., the semantic head of the term).

We think that a similar analysis could be carried out for other languages to provide an assessment of machine learning / deep learning models across different languages.
As for future work, we do believe that progress on language technologies need benchmarks encompassing a variety of tasks in order to favour models that share general linguistic knowledge across tasks. This is very much in the spirit of GLUE, the General Language Understanding Evaluation (Wang et al., 2018), a collection of resources for training, evaluating, and analyzing natural language understanding systems. Our next step will be to collect the Italian resources used in this paper and propose them as a single benchmark for NLP tasks on the Italian language.

10. Bibliographical References

Agirre, E., Cer, D., Diab, M., and Gonzalez-Agirre, A. (2012). SemEval-2012 task 6: A pilot on semantic textual similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393, Montréal, Canada, 7-8 June. Association for Computational Linguistics.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. arXiv.1409.0473 [cs.CL].

Bailoni, T., Dragoni, M., Eccher, C., Guerini, M., and Maimone, R. (2016). Healthy lifestyle support: The PerKApp ontology. In *OWL: Experiences and Directions–Reasoner Evaluation*, pages 15–23. Springer.

Barbieri, F., Basile, V., Croce, D., Nissim, M., Novielli, N., and Patti, V. (2016). Overview of the EVALITA 2016 sentiment polarity classification task. In *Proceedings of EVALITA 2016*, Naples, Italy, December.

Basile, V., Bolioli, A., Patti, V., Rosso, P., and Nissim, M. (2014). Overview of the Evalita 2014 sentiment polarity classification task. In *Proceedings of EVALITA 2014*, Pisa, Italy, December.

Basile, P., Semeraro, G., and Cassotti, P. (2017). Bidirectional LSTM-CNNs-CRF for Italian sequence labeling. In *Proceedings of the Italian Conference on Computational Linguistics (CLiC-it 2017)*, Roma, Italy, December.

Basile, P., Croce, D., Basile, V., and Polignano, M. (2018). Overview of the EVALITA 2018 aspect-based sentiment analysis task (ABSITA). In *Proceedings of EVALITA 2018*, Turin, Italy, December.

Bonadiman, D., Severyn, A., and Moschitti, A. (2015). Deep neural networks for named entity recognition in Italian. In *Proceedings of the Italian Conference on Computational Linguistics (CLiC-it 2015)*, Trento, Italy, December.

Bos, J., Zanzotto, F. M., and Pennacchiotti, M. (2009). Textual entailment at EVALITA 2009. In *Proceedings of EVALITA 2009*, Reggio Emilia, Italy.

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

Craswell, N. (2009). Mean reciprocal rank. *Encyclopedia of Database Systems*, pages 1703–1703.

Dagan, I., Glickman, O., and Magnini, B. (2005). The PASCAL Recognising Textual Entailment Challenge. In *Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*, pages 177–190, Southampton, UK.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv.1810.04805 [cs.CL].

Doddington, G., Mitchell, A., Przybocki, M., Ramshaw, L., Strassel, S., and Weischedel, R. (2004). The automatic content extraction (ACE) program – tasks, data, and evaluation. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal, May. European Language Resources Association (ELRA).

Dolan, W. B. and Brockett., C. (2005). Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005), Asia Federation of Natural Language Processing*.

Dyer, C., Ballesteros, M., Ling, W., Matthews, A., and Smith, N. A. (2015). Transition-based dependency parsing with stack long short-term memory. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 334–343, Beijing, China, July. Association for Computational Linguistics.

Gerevini, A., Lavelli, A., Maffi, A., Maroldi, R., Minard, A.-L. M., Serina, I., and Squassina, G. (2018). Automatic classification of radiological reports for clinical care. *Artificial Intelligence in Medicine*, 91:72 – 81.

Giampiccolo, D., Magnini, B., Dagan, I., and Dolan, B. (2007). The third PASCAL recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 1–9, Prague, June. Association for Computational Linguistics.

Graves, A., Mohamed, A.-r., and Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. IEEE.

Haghighi, A. and Klein, D. (2010). Coreference resolution in a modular, entity-centered model. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 385–393. Association for Computational Linguistics.

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

Kouylekov, M. and Magnini, B. (2005). Recognizing textual entailment with tree edit distance algorithms. In *Proceedings of the First PASCAL Challenges Workshop on Recognising Textual Entailment*, pages 17–20, Southampton, UK.

Kruszewski, G. and Baroni, M. (2015). So similar and yet incompatible: Toward the automated identification of semantically compatible words. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 964–969.

Lafferty, J. D., McCallum, A., and Pereira, F. C. N. (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.

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., and Dyer, C. (2016). Neural architectures for named

entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California, June. Association for Computational Linguistics.

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., and Kang, J. (2019). BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, Sep.

Ma, X. and Hovy, E. (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.

Magnolini, S., Piccioni, V., Balaraman, V., Guerini, M., and Magnini, B. (2019). How to use gazetteers for entity recognition with neural models. In *Proceedings of the 5th Workshop on Semantic Deep Learning (SemDeep-5)*, pages 40–49, Macau, China, 12 August. Association for Computational Linguistics.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Nguyen, T.-V. T., Moschitti, A., and Riccardi, G. (2010). Kernel-based reranking for named-entity extraction. In *Coling 2010: Posters*, pages 901–909, Beijing, China, August. Coling 2010 Organizing Committee.

Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June. Association for Computational Linguistics.

Pianta, E. and Tonelli, S. (2010). Kx: A flexible system for keyphrase extraction. *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 170–173, 01.

Pires, T., Schlinger, E., and Garrette, D. (2019). How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Putelli, L., Gerevini, A. E., Lavelli, A., and Serina, I. (submitted). Deep learning for classification of radiology reports with a hierarchical schema.

Ramshaw, L. and Marcus, M. (1995). Text chunking using transformation-based learning. In *Third Workshop on Very Large Corpora*.

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

Speranza, M. (2009). The named entity recognition task at EVALITA 2009. In *Proceedings of EVALITA 2009*, Reggio Emilia, Italy.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. (2018). GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium, November. Association for Computational Linguistics.

Zanoli, R., Pianta, E., and Giuliano, C. (2009). Named entity recognition through redundancy driven classifiers. In *Proceedings of Evalita 2009*.

11. Language Resource References