IJCoL

Italian Journal of Computational Linguistics

Rivista Italiana di Linguistica Computazionale

Volume 3, Number 2
december 2017

Special Issue:
Natural Language and Learning Machines

aAccademia university press

Registrazione presso il Tribunale di Trento n. 14/16 del 6 luglio 2016

Rivista Semestrale dell'Associazione Italiana di Linguistica Computazionale (AILC)

Associazione Italiana di Linguistica Computazionale

direttore responsabile
Michele Arnese

isbn 978-88-31978-19-4

Accademia University Press
via Carlo Alberto 55
I-10123 Torino
info@aAccademia.it
www.aAccademia.it/IJCoL_3_2

aAccademia university press

IJCoL

Volume 3, Number 2
december 2017

CONTENTS

Introduction to the Special Issue on *Natural Language and Learning Machines*

Dan Roth*
University of Pennsylvania.

Roberto Basili**
Università di Roma, Tor Vergata

1. Introduction

The interaction between machine learning and natural language processing (NLP) research underlies most of the progress made in NLP for the last few decades (Cardie and Mooney 1999; Fung and Roth 2005). Machine Learning has been the common framework for the birth and development of most paradigms, discoveries and achievements in statistical natural language processing. At the international level the AAAI Fall symposiums in 1990 (Jacobs 1990) and 1992 (Goldman 1992) and the IBM TJ Watson paper on statistical Machine Translation (Brown et al. 1988) established firm roots for the use of Bayesian modeling and data-driven algorithms for complex computational linguistic tasks. At that time several Italian research groups were already working on machine learning methods for tasks such as natural language parsing and lexical acquisition. A relevant event was the Workshop *Apprendimento Automatico e Linguaggio Naturale* organized at the University of Torino, whose decisive inspiration was contributed by Leonardo Lesmo and Piero Torasso that pioneered NLP research in Italy ((Lesmo 1997)). One of the topics at the workshop was "Are syntactic representations and parsing still central in current NLP and Information Extraction tasks, given the role that shallow features combined with complex learning algorithms play in achieving significant results over several benchmarks?". As we know, some of these issues and challenges are still relevant today, and these questions still trigger many empirical studies and debates from heterogeneous intellectual positions.

In current research, the aforementioned issues are still open research issues, possibly formulated using a different jargon. Are parsing algorithms still relevant given the growing success demonstrated by recurrent neural networks in tasks that were believed to require parsing? Are linguistic aspects of the problem (e.g. traditional categories such as root vs. lemma distinctions, agreement or verbal aspects) still important given the ability to induce intermediate representations that seems to capture these notions? At the same time one needs to consider ways in which neural networks are currently being trained, mostly counting on vast amounts of task specific annotated data and, consequently, the generality of the representations thus induced.

2. Learning and Language Processing

It has been clearly shown that, in general, (natural language) inference can be formulated as a joint constrained optimization task done over learned components (Roth and

* Department of Computer and Information Science, University of Pennsylvania.
E-mail: danroth@seas.upenn.edu
** Dept. of Enterprise Engineering - Via del Politecnico 1, 00133 Rome, Italy.
E-mail: basili@info.uniroma2.it

Yih 2004, 2007; Chang, Ratinov, and Roth 2012). By “Inference" we refer here to the assignment of values to a collection of interdependent variables. The “optimal" decision model can then be arrived at by satisfying a set of constraints imposed on the final assignment of values to variables, the output decision. For example, the satisfaction of some constraints on the distribution of semantic roles can be jointly optimized with the interpretation of the reference target predicate. As a consequence, learning here corresponds to the learning of (the coefficients of) an objective function that combines a target function ψ (the cost of the proper assignment of values y to variables $\mathcal{Y}$) to be minimized together with the set of theory-driven constraints C, whose individual violations tend to increase the cost, i.e.:

$$y = \text{argmin}_{y \in \mathcal{Y}} w^T \phi(x, y) + u^T C(x, y) \quad (1)$$

where w and u are weights matrices to be learned through annotated data (Roth and Yih 2004; Chang, Ratinov, and Roth 2012), either jointly, or in a decomposed fashion. While $w^T \phi(x, y)$ thus correspond to the decision that the (data-driven) linguistic inference must produce, e.g. semantic role labels y for the individual word sequence x, the $u^T C(x, y)$ component constrains any choice of $y \in \mathcal{Y}$: it is thus helpful in judging the quality of alternative solutions y and ranks them. Joint optimization allows learning to proceed (i.e. carry out the labeling) by maximizing the satisfaction of all constraints.

The above general setting is important in NLP for multiple reasons:

- The decision function corresponds in a more or less direct way to a complex linguistic inference whose nature is in general semantic: it makes a bridge between the observable linguistic symbols x and the operational context (i.e. the world) in which the decision is immersed. For example, the joint assignment of predicate and roles to the incoming sentence x in semantic role labeling.
- The constraints C can be used to express linguistic principles, that embody forms of agreement that natural languages must convey between speakers and hearers. It reflects the expectations one has from an interpretation y of x, whatever the current natural language decision problem (x, y) is. In natural language such agreement is a strongly social phenomena, established across time and possibly through repeated attempts. Constraint optimization just expresses this approximation process.
- Some of the constraints might be derived from the reference world, where properties usually correspond to sound (although simple) theories. These model semantic aspects as well as other formal properties of the decision, are obtained to satisfy external (e.g. domain) knowledge.

The power of natural language results from its variability and its ambiguity. This is also what makes it a highly subjective phenomenon and makes it difficult to process and understand automatically. As is the experience of human subjects, we can say that subjectivity is the ontological status of natural language practices. However, we can foster an objective epistemology of even such highly subjective phenomenon. Machine learning is crucial in this sense. We can say that the increasing success of machine learning in NLP stands as a proof that an epistemologically objective approach to natural language is possible. Machine learning and its mathematics provides sound modeling tools for a vague problem. The constraint optimization model expressed by

Eq. 1 or the convergence properties of the learning algorithms used to model data-driven decisions based on the risk minimization principle are just examples of this contribution. Linguistic inference (as an incremental and iterative agreement process between speakers and hearers) is more easily mapped into a learning (and inference) process that resembles the nature of the language acquisition process. In other words, machine learning, as the ability of machines to develop decision functions, out from examples, and from (being told) constraints, seems a nice way to characterize language processing capabilities as those emerging from linguistic practices.

All the papers collected in this special issue follow, in a more or less tight fashion, the above mathematical setting, although under the umbrella of alternative paradigms, such as deep learning or distributional semantic analysis: they all make strong use of linguistic constraints to control the reference machine learning model. The variety of the tasks and the ways linguistic principles are adopted in the representational hypothesis and in the architectures proposed show the richness of methodologies and open aspects that still inspire research on machine learning for NLP.

3. Overview of the Issue

The first paper by Madotto and Attardi presents a neural network architecture for two tasks, *Reasoning Question Answering* and *Reading Comprehension*. Memory Networks (Weston, Chopra, and Bordes 2014) are employed in order to recognize entities and their relations to answers in a target text. A focus attention mechanism and an independent memory is adopted as an extension of a Recurrent Neural Network. The proposed model, Question Dependent Recurrent Entity Network (QDREN), exploits information and properties of the question during the memorization process and uses them to decide the correctness of one or more proposed answers. The extended network architecture is evaluated on synthetic as well as real datasets with improved accuracy levels and competitive results in both tasks.

In the paper by Passaro and colleagues, a corpus-driven approach to the acquisition of the lexical affective values used in sentiment analysis systems is presented. The acquisition of emotive embeddings for lexical items is realized by co-occurrence analysis with negative expressions. The proposed distributional semantic analysis is a form of bootstrapping for emotional lexicons, built around eight basic emotion categories. In this way, the authors show how to use positive vs. negative lexical valences to model behavioral data.

In the paper by Basile and colleagues presents a complex Deep Learning architecture for the joint learning of several Natural Language Processing tasks for Italian. The architecture is based on state of the art models and exploits both word-level and character-level representations through the integration of Long Short Term Memory (LSTM) networks, Convolutional Neural Networks (CNN) as well as Conditional Random Fields (CRF). The architecture, that provided state of the art performance in several sequence labeling tasks on English datasets, is applied to the Italian language with a multi-task learning paradigm, in particular, targeting PoS-tagging and sentiment analysis. State of the art performance is shown in all the tasks.

In the paper by Bonadiman and colleagues a deep neural network (DNN) for multi-task learning as applied to (three tasks in) the community Question Answering (cQA) process[1]. The latter task, i.e. the new question-old comment similarity estimation, is

1 The CQA process targeted is equivalent to the one proposed in the SemEval-2016 Task 3, i.e., question-comment similarity, question-question similarity and new question-comment similarity.

the task where multi-task learning provides the best contribution. The proposed DNN is jointly trained on all the three cQA tasks and avoids any use of manually designed features and it is shown to approach the state of the art established with methods that make heavy use of feature engineering. It learns to encode questions and comments into a single vector representation shared across the multiple tasks. The results on the official test sets show that the integrated neural network produces higher accuracy and faster convergence rates than the individual one.

... a Closing Remark

The collection of papers in this special issue provides further evidence for the need for stronger and often task specific representations to benefit machine learning for natural language processing. In all papers, complex architectures are obtained either by integrating different learning tasks in one joint training stage or by extending existing architectures. Example of the first approach are the multi-task learning of the individual community Question Answering subproblems in the Bonadiman paper or the joint multi-task learning proposed in Basile and colleagues for POS tagging and sentiment analysis. An example of the second is obtained through the memorization of the input question integrated with multiple sentence embeddings, as proposed by Madotto et al. in the QDREN architecture proposed for Reasoning Question Answering.

The interesting results collected here seem to be all moving in one general direction: the combination of local, i.e. task specific, evidence with general constraints usually derived from a theory of the target linguistic phenomena. As Equation 1 seems to definitively suggest, local (i.e. example specific) constraints should always be combined with theory-driven or expectation-driven constraints (e.g. the attempt to satisfy relational associations between a question and its reference input text). Language studies and linguistic principles thus still seem to have a relevant role in the research towards learning machines that address intelligence.

References

Brown, Peter F., John Cocke, Stephen Della Pietra, Vincent J. Della Pietra, Frederick Jelinek, Robert L. Mercer, and Paul S. Roossin. 1988. A statistical approach to language translation. In *Proceedings of the 12th International Conference on Computational Linguistics, COLING '88, Budapest, Hungary, August 22-27, 1988*, pages 71–76.

Cardie, Claire and Raymond J. Mooney. 1999. Guest editors; introduction: Machine learning and natural language. *Mach. Learn.*, 34(1-3):5–9, February.

Chang, Ming-Wei, Lev Ratinov, and Dan Roth. 2012. Structured learning with constrained conditional models. *Machine Learning*.

Fung, Pascale and Dan Roth. 2005. Guest editors introduction: Machine learning in speech and language technologies. *Machine Learning*, 60(1-3):5–9.

Goldman, Robert. 1992. *Working notes of the 1992 AAAI Fall Symposium "Intelligent Probabilistic Approaches to Natural Language"*. AAAI Press, Menlo Parl, California, USA.

Jacobs, P.S:. 1990. *Working notes, 1990 AAAI Spring Symposium on Text-Based Intelligent Systems, appeared as Text-based intelligent systems: current research in text analysis, information extraction, and retrieval, Report 90CRD198*. General Electric R. & D. Centre, Schenectady, NY.

Lesmo, Leonardo. 1997. *Incontro dei Gruppi di Lavoro dell'Associazione Italiana per l'Intelligenza Artificiale su Apprendimento Automatico e Linguaggio Naturale*. Universita' di Torino, Torino.

Roth, D. and W. Yih. 2007. Global inference for entity and relation identification via a linear programming formulation. *Introduction to Statistical Relational Learning. Editors: Lise Getoor and Ben Taskar, 2007*.

Roth, Dan and Wen-tau Yih. 2004. A linear programming formulation for global inference in natural language tasks. In *CoNLL 2004*, pages 1–8.

Weston, Jason, Sumit Chopra, and Antoine Bordes. 2014. Memory networks. *CoRR*, abs/1410.3916.

Question Dependent Recurrent Entity Network for Question Answering

Andrea Madotto* **
The Hong Kong University of Science and Technology

Giuseppe Attardi†
University of Pisa

Question Answering is a task which requires building models capable of providing answers to questions expressed in human language. Full question answering involves some form of reasoning ability. We introduce a neural network architecture for this task, which is a form of Memory Network, that recognizes entities and their relations to answers through a focus attention mechanism. Our model is named Question Dependent Recurrent Entity Network and extends the Recurrent Entity Network by exploiting aspects of the question during the memorization process. We validate the model on both synthetic and real datasets: the bAbI question answering dataset and the CNN & Daily News reading comprehension dataset. In our experiments, our models improved the existing Recurrent Entity Network and achieved competitive results in both dataset.

1. Introduction

Question Answering is a task that requires capabilities beyond simple Natural Language Processing since it involves both linguistic techniques and inference abilities. Both the document sources and the questions are expressed in natural language, which is ambiguous and complex to understand. To perform such a task, a model needs in fact to understand the underlying meaning of a text. Achieving this ability is quite challenging for a machine since it requires a reasoning phase (chaining facts, basic deductions, etc.) over knowledge extracted from the plain input data. In this article, we focus on two Question Answering tasks: a *Reasoning Question Answering* (RQA) and a *Reading Comprehension* (RC). These tasks are tested by submitting questions to be answered directly after reading a piece of text (e.g. a document or a paragraph).

Recent progress in the field has been possible thanks to machine learning algorithms which automatically learn from large collections of data. Deep Learning (LeCun, Bengio, and Hinton 2015) algorithms achieve the current State-of-The-Art in our tasks of interest. A particularly promising approach is based on *Memory Augmented Neural Networks*. These networks are also known as *Memory Networks* (Weston, Chopra, and Bordes 2015) or *Neural Turing Machines* (Graves, Wayne, and Danihelka 2014). In the literature the RQA and RC tasks are typically solved by different models. However, the two tasks share a similar scope and structure. We propose to tackle both with a model

* Human Language Technology Center, Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong.
E-mail: eeandreamad@ust.hk
** Work done while at University of Pisa
† Dipartimento di Informatica, University of Pisa, Largo B. Pontecorvo, 3. E-mail: attardi@di.unipi.it

called *Question Dependent Recurrent Entity Network* (QDREN), which improves over the model called *Recurrent Entity Network* (Henaff et al. 2017).

Our major contributions are: 1) exploiting knowledge of the question for storing relevant facts in memory, 2) adding a tighter regularization scheme, and 3) changing the activation functions. We test and compare our model on two datasets, bAbI (Weston et al. 2016) and CNN & Daily News (Hermann et al. 2015), which are standard benchmark for both tasks. The rest of the paper is organized as follows: section *Related* outlines the models used in QA tasks, while section *Model* discusses the proposed QDREN model. Section *Experiments and Results* shows training details and performance achieved by our model. The section *Analysis* reports a visualization with the aim to explain the obtained results. Finally, section *Conclusions* summarizes the work done.

2. Related Work

2.1 Reasoning Question Answering

A set of synthetic tasks, called bAbI (Weston et al. 2016), has been proposed for testing the ability of a machine in chaining facts, performing simple inductions or deductions. These tasks became a standard benchmark for measuring reasoning QA, several examples are shown in Table 1 [1]. The dataset is available in two sizes, 1K and 10K training samples, and in two settings, i.e. with and without supporting facts. The latter allows knowing which facts in the input are needed for answering the question (i.e. a stronger supervision). Obviously, the 1K sample setting with no supporting facts is quite hard to handle, and it is still an open research problem. *Memory Network* (Weston, Chopra, and Bordes 2015) was one of the first models to provide the ability to explicitly store facts in memory, achieving good results on the bAbI dataset. An evolution of this model is the *End to End Memory Network* (Sukhbaatar et al. 2015), which allows for end-to-end training. This model represents the State-of-The-Art in the bAbI task with 1K training samples. Several other models have been tested in the bAbI tasks achieving competitive results, such as *Neural Turing Machine* (Graves, Wayne, and Danihelka 2014), *Differentiable Neural Computer* (Graves et al. 2016) and *Dynamic Memory Network* (Kumar et al. 2015, Xiong, Merity, and Socher 2016). Several other baselines have also been proposed (Weston et al. 2016), such as: an n-gram (Richardson, Burges, and Renshaw 2013) models, an LSTM reader and an SVM model. However, some of them still required strong supervision by means of the supporting facts.

2.2 Reading Comprehension

Reading Comprehension is defined as the ability to read some text, process it, and understand its meaning. A impending issue for tackling this task was to find suitably large datasets with human annotated samples. This shortcoming has been addressed by collecting documents which contain easy recognizable short summary, e.g. news articles, which contain a number of bullet points, summarizing aspects of the information contained in the article. Each of these short summaries is turned into a fill-in question template, by selecting an entity and replacing it with an anonymized placeholder.

Three datasets follows this style of annotation: *Children's Text Books* (Hill et al. 2016), *CNN & Daily Mail news articles* (Hermann et al. 2015), and *Who did What* (Onishi et al.

1 Interested readers can find all the tasks examples in (Weston et al. 2016)

Table 1
bAbI dataset examples.

Task	*Counting*	*Lists/Sets*
Story	Daniel picked up the football. Daniel dropped the football. Daniel got the milk. Daniel took the apple.	Daniel picks up the football. Daniel drops the newspaper. Daniel picks up the milk. John took the apple.
Question	How many objects is Daniel holding?	What is Daniel holding?
Answer	Two	Milk, football
Task	*Three Argument Relations*	*Yes/No Questions*
Story	Mary gave the cake to Fred. Fred gave the cake to Bill. Jeff was given the milk by Bill.	John moved to the playground. Daniel went to the bathroom. John went back to the hallway.
Question	Who gave the cake to Fred?	Is John in the playground?
Answer	Mary	No

2016). It is also worth to mention *Squad* (Rajpurkar et al. 2016), a human annotated dataset from Stanford NLP group. *Memory Networks*, described in the previous subsection, has been tested (Hill et al. 2016) on both the CNN and CBT datasets, achieving good results. The *Attentive and Impatient Reader* (Hermann et al. 2015) was the first model proposed for the *CNN & Daily Mail* dataset, and it is therefore often used as a baseline. While this model achieved good initial results, shortly later a small variation to such model, called *Standford Attentive Reader* (Chen, Bolton, and Manning 2016), increased its accuracy by 10%. Another group of models are based on an Artificial Neural Network architecture called *Pointer Network* (Vinyals, Fortunato, and Jaitly 2015). *Attentive Sum Reader* (Kadlec et al. 2016) and *Attention over Attention* (Cui et al. 2017) use a similar idea for solving different reading comprehension tasks. *EpiReader* (Trischler et al. 2016) and *Dynamic Entity Representation* (Kobayashi et al. 2016), partially follow the *Pointer Network* framework but they also achieve impressive results in the RC tasks. Also for this task several baselines, both learning and non-learning, have been proposed. The most commonly used are: *Frame-Semantics*, Word distance, and *LSTM Reader* (Hermann et al. 2015) and its variation (windowing etc.).

3. Proposed Model

Our model is based on the *Recurrent Entity Network* (REN) (Henaff et al. 2017) model. The latter is the only model able to pass all the 20 bAbI tasks using the 10K sample size and without any supporting facts. However, this model fails many tasks with the 1K setting, and it has not been tried on more challenging RC datasets, like the CNN news articles. Thus, we propose a variant to the original model called *Question Dependent Recurrent Entity Network* ($QDREN$)[2]. This model tries to overcome the limitations of

2 An implementation is available at https://github.com/andreamad8/QDREN

the previous approach. The model consists in three main components: *Input Encoder*, *Dynamic Memory*, and *Output Module*.

The training data consists of tuples $\{(x_i, y_i)\}_{i=1}^{n}$, with n equal to the sample size, where: x_i is composed by a tuple (T, q), where T is a set of sentences $\{s_1, \ldots, s_t\}$, each of which has a maximum of m words, and q a single sentence with k words representing the question. Instead, y_i is a single word that represents the answer.

The Input Encoder transforms the set of words of a sentence s_t and the question q into a single vector representation by using a multiplicative mask. Let's define $E \in \mathbb{R}^{|V| \times d}$ the embedding matrix[3], that is used to convert words to vectors, i.e. $E(w) = e \in \mathbb{R}^d$. Hence, $\{e_1, \ldots, e_m\}$ are the word embedding of each word in the sentence s_t and $\{e_1, \ldots, e_k\}$ the embedding of the question's words. The multiplicative masks for the sentences are defined as $f^{(s)} = \{f_1^{(s)}, \ldots, f_k^{(s)}\}$ and $f^{(q)} = \{f_1^{(q)}, \ldots, f_k^{(q)}\}$ for the question, where each $f_i \in \mathbb{R}^d$. The encoded vector of a sentence is defined as:

$$s_t = \sum_{r=1}^{m} e_r \odot f_r^{(s)} \qquad\qquad q = \sum_{r=1}^{k} e_r \odot f_r^{(q)}$$

Dynamic Memory stores information of entities present in T. This module is very similar to a Gated Recurrent Unit (GRU) (Cho et al. 2014) with a hidden state divided into blocks. Moreover, each block ideally represents an entity (i.e. person, location etc.), and it stores relevant facts about it. Different datasets may require different number of blocks, in the experiment section we will further discuss this issue. Each block i is made of a hidden state $h_i \in \mathbb{R}^d$ and a key $k_i \in \mathbb{R}^d$, where d is the embedding size. The Dynamic Memory module is made of a set of blocks, which can be represent with a set of hidden states $\{h_1, \ldots, h_z\}$ and their correspondent set of keys $\{k_1, \ldots, k_z\}$. The equation used to update a generic block i are the following:

$$\begin{aligned}
g_i^{(t)} &= \sigma(s_t^T h_i^{(t-1)} + s_t^T k_i^{(t-1)} + s_t^T q) && \text{(Gating Function)} \\
\hat{h}_i^{(t)} &= \phi(U h_i^{(t-1)} + V k_i^{(t-1)} + W s_t) && \text{(Candidate Memory)} \\
h_i^{(t)} &= h_i^{(t-1)} + g_i^{(t)} \odot \hat{h}_i^{(t)} && \text{(New Memory)} \\
h_i^{(t)} &= h_i^{(t)} / \|h_i^{(t)}\| && \text{(Reset Memory)}
\end{aligned}$$

where σ represents the sigmoid function, ϕ a generic activation function which can be chosen among a set (e.g. sigmoid, ReLU, etc.). $g_i^{(t)}$ is the gating function which determines how much the ith memory should be updated, and $\hat{h}_i^{(t)}$ is the new candidate value of the memory to be combined with the existing one $h_i^{(t-1)}$. The matrix $U \in \mathbb{R}^{d \times d}$, $V \in \mathbb{R}^{d \times d}$, $W \in \mathbb{R}^{d \times d}$ are **shared** among different blocks, and are trained together with the key vectors. The addition of the $s_t^T q$ term in the gating function is our main contribution. We add such term with the assumption that the question can be useful to focus the attention of the model while analyzing the input sentences.

3 Where $|V|$ is the vocabulary size and d the embedding dimension.

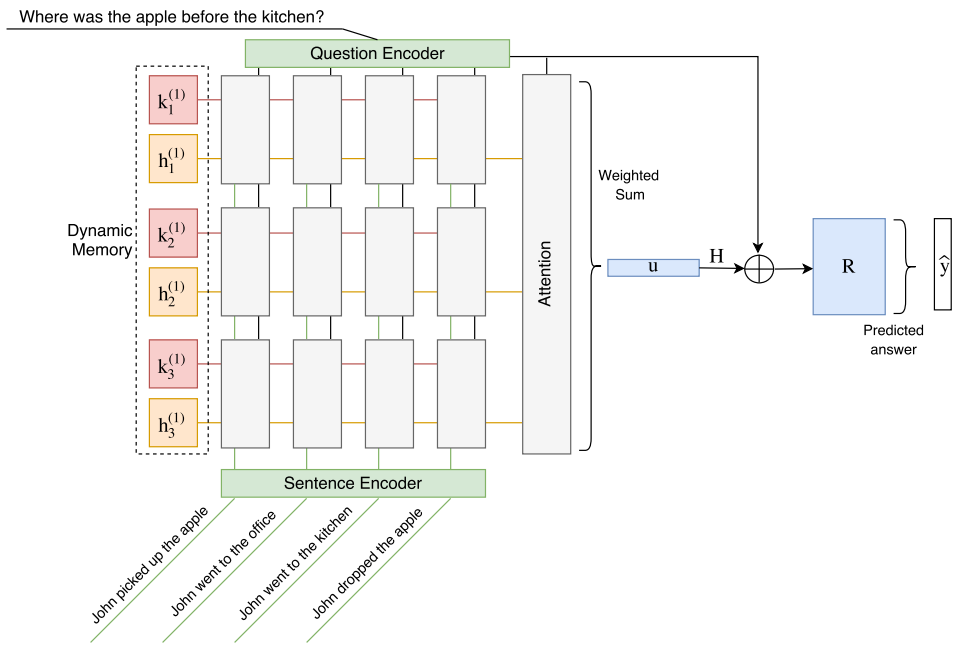


Figure 1
Conceptual schema of the QDREN model, with three memory blocks. In input a sample taken from bAbI task dataset.

The Output Module creates a probability distribution over the memories' hidden states using the question q. Thus, the hidden states are summed up, using the probability as weight, to obtain a single state representing all the input. Finally, the network output is obtained by combining the final state with the question. Let us define $R \in \mathbb{R}^{|V| \times d}$, $H \in \mathbb{R}^{d \times d}$, $\hat{y} \in \mathbb{R}^{|V|}$, z is the number of blocks, and ϕ can be chosen among different activation functions. Then, we have:

$$
\begin{aligned}
p_i &= \text{Softmax}(q^T h_i) \\
u &= \sum_{j=1}^{z} p_j h_j \\
\hat{y} &= R\phi(q + Hu)
\end{aligned}
$$

The model is trained using a cross entropy loss $H(\hat{y}, y)$ plus L2 regularisation term, where y is the one hot encoding of the correct answer. The sigmoid function and the L2 term are two novelty added to the original REN. Overall, the trainable parameters are:

$$\Theta = [E, f^{(s)}, f^{(q)}, U, V, W, k_1, \ldots, k_z, R, H]$$

where $f^{(s)}$ refers to the sentence multiplicative masks, $f^{(q)}$ to the question multiplicative masks, and each k_i to the key of a generic block i. The number of parameters is dominated by E and R, since they depend on the vocabulary size. However, R is normally

is much smaller than E like in the CNN dataset, in which the prediction is made on a restricted number of entities[4]. All the parameters are learned using the Backpropagation Through Time (BPTT) algorithm. A schematic representation of the model is shown in Figure 1.

4. Experiments and Results

Our model has been implemented using TensorFlow v1.1 (Abadi et al. 2015) and the experiments have been run on a Linux server with 4 Nvidia P100 GPUs. As mentioned earlier, we tested our model in two datasets: the bAbI 1k sample and the CNN news articles. The first dataset have 20 separate tasks, each of which has 900/100/1000 training, validation, and test samples. Instead, the second one has 380298/3924/3198 training, validation and test samples. We kept the original splitting to compare our results with the existing ones.

bAbI:. in these tasks, we fixed the batch size to 32, we did not use any pre-trained word embedding, and we used Adam (Kingma and Ba 2015) optimizer. In all the experiment we used 20 blocks of memory since it is equal to the maximum number of entity in each story. We have also clipped the gradient to a maximum of 40 (to avoid gradient explosion), and we set the word embedding size to 100, as it has also been suggested in the original paper. We have also implemented an early stopping method, which stop the training ones the validation accuracy does not improve after 50 epochs. Several values for the hyper-parameter have been tried and, for each task, we selected the setting that achieved the highest accuracy in validation. Once we selected the best model, we estimate its generalization error using the provided Test set. Table 2 shows an example of the dataset and the used hyper-parameters. We compared our results

Table 2
On the left an example of the bAbI task, and on the right the selected model hyper-parameters.

Story	**Question**
John picked up the apple John went to the office John went to the kitchen John dropped the apple	Where was the apple before the kitchen?
	Answer
	office

Parameter	**Values**
Learning Rate (α)	0.01,0.001,0.0001
Number of Blocks	20,30,40,50
L2 reg. (λ)	0,0.001,0.0001
Dropout (**Dr**)	0.3,0.5,0.7

with four models: n-gram model, LSTM, original REN (with no question in the gating function) and *End To End Memory Network* (MemN2N) (Sukhbaatar et al. 2015), which is currently the State-Of-The-Art in this setting. To the best of our knowledge we achieved the lowest number of failed tasks, failing just 8 tasks compared with the previous State-Of-The-Art which was 11. Comparing our QDREN with the original *Recurrent Entity Network* (REN) we achieved, on average, an improvement of 11% in the average error rate and we passed 7 tasks more. Table 3 shows the error rate[5] in the test set obtained using each compared model, and the hyper-parameter setting used in each task. We improve the mean error compared to the original REN, however we still do know reach the error rate achieved by the *End To End Memory Network* (even if we passed more

4 Therefore $R \in \mathbb{R}^{|entities| \times d}$
5 The error is the percentage of wrong answers.

Table 3
Test set error rate comparison between n-gram, LSTM, QDREN, REN and End To End Memory Network (MemN2N). All the results have been taken from the original articles. In bold we highlight the task in which we greatly outperform the other models. On the right the hyper-parameters used in QDREN.

Task	n-gram	LSTM	MemN2N	REN	QDREN	Blk	λ	α	Dr
1	64	50	0	0.7	0	20	0	0.001	0.5
2	98	80	8.3	56.4	67.6	30	0	0.001	0.5
3	93	80	40.3	69.7	60.8	40	0	0.001	0.5
4	50	39	2.8	1.4	0	20	0	0.001	0.5
5	80	30	13.1	4.6	**2.0**	50	0	0.001	0.2
6	51	52	7.6	30	29	30	0	0.001	0.5
7	48	51	17.3	22.3	**0.7**	30	0	0.001	0.5
8	60	55	10	19.2	**2.5**	20	0.001	0.001	0.7
9	38	36	13.2	31.5	**4.8**	40	0.0001	0.001	0.5
10	55	56	15.1	15.6	**3.8**	20	0	0.001	0.5
11	71	28	0.9	8	**0.6**	20	0	0.001	0.5
12	91	26	0.2	0.8	0	20	0	0.0001	0.5
13	74	6	0.4	9	**0.0**	40	0.001	0.001	0.7
14	81	73	1.7	62.9	15.8	30	0.0001	0.001	0.5
15	80	79	0	57.8	**0.3**	20	0	0.001	0.5
16	57	77	1.3	53.2	52	20	0.001	0.001	0.5
17	54	49	51	46.4	37.4	40	0.001	0.001	0.5
18	48	48	11.1	8.8	10.1	30	0.0001	0.001	0.5
19	10	92	82.8	90.4	85	20	0	0.001	0.5
20	24	9	0	2.6	0.2	20	0	0.001	0.5
Failed Tasks (>5%):	20	20	11	15	8				
Mean Error:	65.9	50.8	13.9	29.6	18.6				

tasks). It is worth to notice the following two facts: first, in task 14 and 18 the error is very close to the threshold for passing the task (5%); second, in task 2, we achieved a slightly worse result (10% error more) with respect to the original REN.

CNN news articles:. in this dataset, the entities in the original paragraph are replaced by an ID, making the task even more challenging. The CNN dataset is already tokenized and cleaned, therefore we did not apply any text pre-processing. As it was done in other models, the set of possible answers is restricted to the set of hidden entities in the text, that are much less, around 500, compared to all the words (120K) in the vocabulary. Compared to the model used for bAbI, we changed the activation function of the output layer, using a sigmoid instead of parametric ReLU, since after several experiments we noticed that such activation was hurting the model performance. Moreover, the input was not split into sentences, thus we divided the text into sentences using the dot token ("."). sentence splitting in general is itself a challenging task, but in this case the input was already cleaned and normalised. However, the sentence may be very long, thus we introduced a windowing mechanism. The same approach has been used in the *End To End Memory Network* (Sukhbaatar et al. 2015) as a way to encode the input sentence. This method takes each entity marker ($@entity_i$) and it creates a window of b words around it. Formally, $\{w_{i-\frac{(b-1)}{2}}, \ldots, w_i, \ldots, w_{i+\frac{(b-1)}{2}}\}$, where w_i represent the entity of interest. For the question, a single window is created around the placeholder marker (the word to predict). Moreover, we add $2(b-1)$ tokens for the entities at the beginning and at the end of the text. To check whether our QDREN could improve the existent REN and whether the window-based approach makes any difference in comparison with plain sentences, we separately trained four different models: REN+SENT, REN+WIND, QDREN+SENT and QDREN+WIND. Where SENT represent simple input sentences,

and WIND the window as a input. For each of this model, we conduct a separated model selection using a various number of hyper-parameters. Table 4 shows an example of the dataset and the used hyper-parameters. As for the bAbI task, we used early

Table 4
On the left, an example from CNN news article, and on the right, the model selection Hyper-parameters.

Story	**Question**
(@entity1) @entity0 may be @entity2 in the popular @entity4 superhero films but he recently dealt in some advanced bionic technology ...	"@placeholder" star @entity0 presents a young child
	Answer
	@entity2

Parameter	**Values**
Learning Rate (α)	0.1,0.01,0.001,0.0001
Window	2,3,4,5
Number of Blocks	10,20,50,70,90
L2 reg. (λ)	0.0,0.001,0.0001,0.00001
Optimizer	Adam,RMSProp
Batch Size	128,64,32
Dropout (**Dr**)	0.2,0.5,0.7,0.9

stopping, ending the training once the validation accuracy does not improve for 20 epochs. Since each training required a large amounts of time (using a batch size of 64 an epoch takes around 7 hours), we opted for a random search technique (Bergstra and Bengio 2012), and we used just a sub-sample of the training set, i.e. 10K sample, for the model selection, but we still keep the validation set as it was. Obviously, this is not an optimal parameter tuning, since the model is selected on just 10K samples. Indeed, we noticed that the selected model, which is trained using all the samples (380K), tends to under-fit. However, it was the only way to try different parameters in a reasonable amount of time. Moreover, we also limited the vocabulary size to the most common 50K words, and we initialize the embedding matrix using Glove (Pennington, Socher, and Manning 2014) pre-trained word embedding of size 100. As for bAbI, we used 20 blocks of memory. As before, we selected the models that achieved the highest accuracy

Table 5
Test set accuracy comparison between **REN+SENT**, **QDREN+SENT**, **REN+WIND** and **QDREN+WIND**. We show the best hyper-parameters picked by the model selection, and the accuracy values.

	REN+SENT	**QDREN+SENT**	**REN+WIND**	**QDREN+WIND**
Number of Blocks	20	10	50	20
Window	-	-	5	4
Learning Rate	0.001	0.001	0.0001	0.01
Optimizer	Adam	Adam	RMSProp	RMSProp
Dropout	0.7	0.2	0.5	0.5
Batch Size	128	64	64	64
λ	0.0001	0.001	0.0001	0.0001
Loss Training	2.235	2.682	2.598	2.216
Loss Validation	2.204	2.481	2.427	1.885
Loss Test	2.135	2.417	2.319	1.724
Accuracy Training	0.418	0.349	0.348	0.499
Accuracy Validation	0.420	0.399	0.380	0.591
Accuracy Test	0.420	0.397	0.401	**0.628**

in the validation set, and then we estimate its generalization error using the provided test set. The selected models, with their hyper-parameters, are shown in Table 5. The

Table 6
Validation/Test accuracy (%) on CNN dataset. All the reported results are taken from the original articles: Max Freq.,Frame-semantic, Attentive Reader, Word distance, Impatient Reader, LSTM Reader from (Hermann et al. 2015), Attentive Reader(Chen, Bolton, and Manning 2016), AS for Attentive Sum(Kadlec et al. 2016), AoA for Attention over Attention (Cui et al. 2017), and DER for Dynamic Entity Representation(Kobayashi et al. 2016).

	Val	*Test*		*Val*	*Test*		*Val*	*Test*
Max Freq.	30.5	33.2	**MemN2N**	63.4	66.8	**AS Reader**	68.6	69.5
Frame-semantic	36.3	40.2	**Attentive Reader**	61.6	63	**AoA**	73.1	**74.4**
Word distance	50.5	50.9	**Impatient Reader**	61.8	63.8	**EpiReader**	73.4	74
LSTM Reader	55	57	**Stanford (AR)**	72.5	72.7	**DER**	71.3	72.9

best accuracy[6] is achieved by QDREN+WIND with a value of 0.628, while all other models could not achieve an accuracy greater than 0.42. The window-based version without question supervision could not achieve an accuracy higher than 0.401. Indeed, saving only facts relative to the question seems to be the key to achieving a good score in this task. We also noticed that using plain sentences, even with QDREN, we cannot achieve a higher accuracy. This might be due to the sentence encoder, since just using the multiplicative masks does not provide enough expressive power for getting key features of the sentence. Moreover, we notice that the accuracy achieved in the training set is always lower than that in the validation and test set. The same phenomenon is present also in other models, in our particular case this might be due to the strong regularization term used in our models. Our model achieves an accuracy comparable to the *Attentive and Impatient Reader* (Hermann et al. 2015), but not yet State-Of-The-Art model (i.e. *Attention over Attention* (AoA)). It is worth noting though that our model is much simpler and it goes through each paragraph just once. A summary of the other models' results are shown in Table 6.

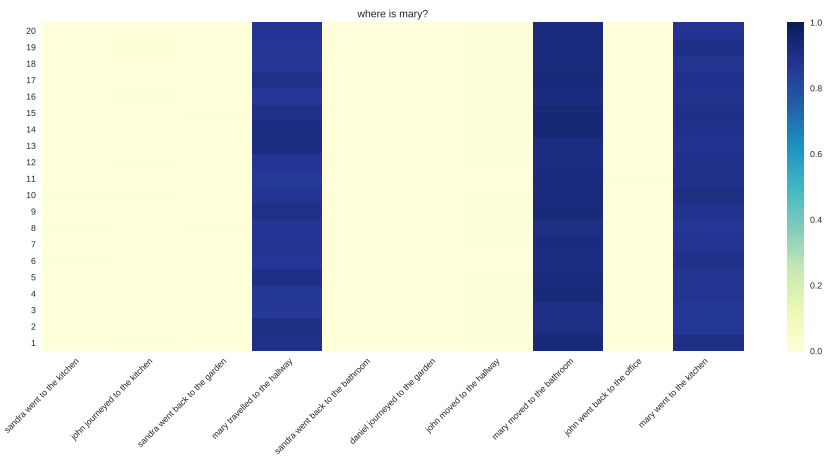
5. Analysis

To better understand how our proposed model (i.e. QDREN) works and how it improves the accuracy of the existing REN, we studied the gating function behavior. Indeed, the output of this function decides how much and what we store in each memory cell, and it is also where our model differs from the original one. Moreover, we trained the QDREN and the original REN using the bAbI task number 1 using 20 memory blocks. The latter mention number of blocks has been selected heuristically by knowing that the maximum number of entity in the facts is 20. We pick up this task since both models pass it, and it is one of the simplest, which also allows to better understand and visualize the results. Indeed, we have tried to visualize other tasks but the result was difficult to understand since there were too many sentences in input and it was difficult to understand how the gate opened. The visualization result is shown in Figure 2, where we plotted the activation matrix for both models, using a sample of the validation set. In these plots, we can notice how the two models learn which information to store.

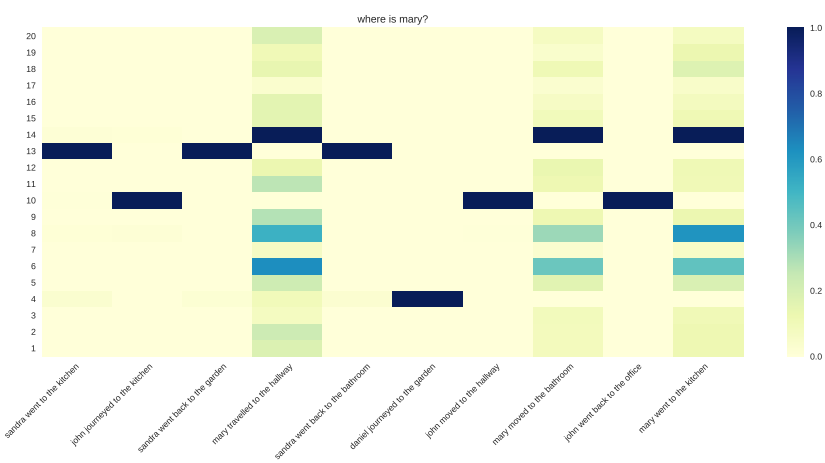
In Figure 2 (a), we notice that the QDREN is opening the gates just when in the

6 Percentage of correct answers.

sentence appears the entity named Mary. This because such entity is also present in the question (i.e., "where is Mary?"). Even though the model is focusing on the right entity, its gates are opening all at the same times. In fact, we guess that a sparser activation would be better since it may have modeled which other entities are relevant for the final answer. Instead, the gaiting activation of the original REN is sparse, which is good if we would like to learn all the relevant facts in the text. Indeed, the model effectively assigns a block to each entity and it opens the gates just ones such entity appears in the input sentences. For example, in Figure 2 (b) the block cell number 13 supposedly



(a)



(b)

Figure 2
Heatmap representing the gating function result for each memory block. In the y-axes represents the memory block number (20 in this example), in the x-axes, there are the sentences in the input divided into time steps, and at the top, there is the question to be answered. Darker color means a gate more open (values close to 1) and lighter colour means the gate less open. (a) shows QDREN and (b) shows REN.

represents the entity Sandra, since each sentence in which this name appears the gate function of the block fully opens (value almost 1). Futher, we can notice the same phenomenon with the entity John (cell 10), Daniel (cell 4), and Mary (cell 14). Other entities (e.g., kitchen, bathroom, etc.) are more difficult to recognize in the plot since their activation is less strong and probably distributes this information among blocks.

6. Conclusion

In this paper we presented the *Question Dependent Recurrent Entity Network*, used for reasoning and reading comprehension tasks. This model uses a particular RNN cell in order to store just relevant information about the given question. In this way, in combination with the original *Recurrent Entity Network* (keys and memory), we improved the success rate in the bAbI 1k task and achieved promising results in the Reading comprehension task on the *CNN & Daily news* dataset. However, we believe that there are still margins for improving the behavior for the proposed cell. Indeed, the cell has not enough expressive power to make a selective activation among different memory blocks (notice in Figure 2 (a) the gates open for all the memories).

Acknowledgments

This work has been supported in part by grant no. GA_2016_009 "Grandi Attrezzature 2016" by the University of Pisa.

References

Abadi, Martín, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Bergstra, James and Yoshua Bengio. 2012. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305.

Chen, Danqi, Jason Bolton, and Christopher D. Manning. 2016. A thorough examination of the cnn/daily mail reading comprehension task. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2358–2367, Berlin, Germany. Association for Computational Linguistics.

Cho, Kyunghyun, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar. Association for Computational Linguistics.

Cui, Yiming, Zhipeng Chen, Si Wei, Shijin Wang, Ting Liu, and Guoping Hu. 2017. Attention-over-attention neural networks for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 593–602, Vancouver, Canada. Association for Computational Linguistics.

Graves, Alex, Greg Wayne, and Ivo Danihelka. 2014. Neural turing machines. *CoRR*.

Graves, Alex, Greg Wayne, Malcolm Reynolds, Tim Harley, Ivo Danihelka, Agnieszka Grabska-Barwińska, Sergio Gómez Colmenarejo, Edward Grefenstette, Tiago Ramalho, John Agapiou, et al. 2016. Hybrid computing using a neural network with dynamic external memory. *Nature*, 538(7626):471–476.

Henaff, Mikael, Jason Weston, Arthur Szlam, Antoine Bordes, and Yann LeCun. 2017. Tracking the world state with recurrent entity networks. *International Conference on Learning*

Representations.
Hermann, Karl Moritz, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, pages 1693–1701, Cambridge, MA, USA. MIT Press.
Hill, Felix, Antoine Bordes, Sumit Chopra, and Jason Weston. 2016. The goldilocks principle: Reading children's books with explicit memory representations. *International Conference on Learning Representations*.
Kadlec, Rudolf, Martin Schmid, Ondřej Bajgar, and Jan Kleindienst. 2016. Text understanding with the attention sum reader network. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 908–918, Berlin, Germany. Association for Computational Linguistics.
Kingma, Diederik and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *International Conference for Learning Representations*.
Kobayashi, Sosuke, Ran Tian, Naoaki Okazaki, and Kentaro Inui. 2016. Dynamic entity representation with max-pooling improves machine reading. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 850–855, San Diego, California, USA. Association for Computational Linguistics.
Kumar, Ankit, Ozan Irsoy, Jonathan Su, James Bradbury, Robert English, Brian Pierce, Peter Ondruska, Ishaan Gulrajani, and Richard Socher. 2015. Ask me anything: Dynamic memory networks for natural language processing. *CoRR, abs/1506.07285*.
LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature*, 521(7553):436–444.
Onishi, Takeshi, Hai Wang, Mohit Bansal, Kevin Gimpel, and David McAllester. 2016. Who did what: A large-scale person-centered cloze dataset. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2230–2235, Austin, Texas, USA. Association for Computational Linguistics.
Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar.
Rajpurkar, Pranav, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas, USA. Association for Computational Linguistics.
Richardson, Matthew, Christopher J.C. Burges, and Erin Renshaw. 2013. Mctest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 193–203, Seattle, USA. Association for Computational Linguistics.
Sukhbaatar, Sainbayar, Arthur Szlam, Jason Weston, and Rob Fergus. 2015. End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448.
Trischler, Adam, Zheng Ye, Xingdi Yuan, Philip Bachman, Alessandro Sordoni, and Kaheer Suleman. 2016. Natural language comprehension with the epireader. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 128–137, Austin, Texas, USA. Association for Computational Linguistics.
Vinyals, Oriol, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 28*. Curran Associates, Inc., pages 2692–2700.
Weston, Jason, Antoine Bordes, Sumit Chopra, Alexander M. Rush, Bart van Merriënboer, Armand Joulin, and Tomas Mikolov. 2016. Towards ai-complete question answering: A set of prerequisite toy tasks. *International Conference on Learning Representations*.
Weston, Jason, Sumit Chopra, and Antoine Bordes. 2015. Memory networks. *International Conference on Learning Representations*.
Xiong, Caiming, Stephen Merity, and Richard Socher. 2016. Dynamic memory networks for visual and textual question answering. *CoRR*, 1603.

Learning Affect with Distributional Semantic Models

Lucia C. Passaro*
Università di Pisa

Alessandro Bondielli**
Università degli Studi di Firenze

Alessandro Lenci†
Università di Pisa

The affective content of a text depends on the valence and emotion values of its words. At the same time a word distributional properties deeply influence its affective content. For instance a word may become negatively loaded because it tends to co-occur with other negative expressions. Lexical affective values are used as features in sentiment analysis systems and are typically estimated with hand-made resources (e.g. WordNet Affect), which have a limited coverage. In this paper we show how distributional semantic models can effectively be used to bootstrap emotive embeddings for Italian words and then compute affective scores with respect to eight basic emotions. We also show how these emotive scores can be used to learn the positive vs. negative valence of words and model behavioral data.

1. Introduction

In recent years, cognitive science and computational linguistics have seen a rising interest in subjectivity, opinions, feelings and emotions. In psycholinguistics, *valence*, *arousal* and *dominance* are considered the three main dimensions to measure the emotional value of a word. Warriner, Kuperman, and Brysbaert (2013) define these dimensions as follows. Valence is "pleasantness of the stimulus", usually ranging from 1 (very unpleasant) to 9 (very pleasant). An example of a word with low valence is *dead*, whereas *holiday* has an high value. Arousal is instead the *intensity* of the feeling evoked, on a scale from "stimulated" to "unaroused". *Passion* is a highly arousing word, whilst *sleep* is not arousing. Finally, dominance is identified as the degree of "control" felt by a reader given the word as stimulus (Louwerse and Recchia 2014). For example, *victory* is a word with a very high dominance rating. In computational linguistics, the goal moves from the investigation of such psycholinguistic variables at the lexical level to the classification of texts with respect to the emotions they express or, in the case of Sentiment Analysis, to their affective valence.

It is clear that these research areas are closely interrelated, but unfortunately they often tend to ignore each other and to use different methods to create, extend and evaluate their resources. The aim of this work is to show how distributional semantic models can

* Computational Linguistics Laboratory, Dipartimento di Filologia, Letteratura e Linguistica (FiLeLi) - Via S. Maria 36, 56126 Pisa, Italy E-mail: lucia.passaro@fileli.unipi.it
** Dipartimento di Ingegneria dell'Informazione (DINFO) - Via Santa Marta 3, 50139 Firenze, Italy E-mail: alessandro.bondielli@unifi.it
† Computational Linguistics Laboratory, Dipartimento di Filologia, Letteratura e Linguistica (FiLeLi) - Via S. Maria 36, 56126 Pisa, Italy E-mail: alessandro.lenci@unipi.it

be used to bootstrap emotive *embeddings* for Italian words and then compute affective scores with respect to eight basic emotions. We also show how these emotive scores can be used to learn the positive vs. negative valence of words to model behavioral data. We will test the results on human-based ratings, assuming that the rated valence can be defined as the "polarity of emotional activation" (Lang, Bradley, and Cuthbert 1997).

One possible approach to infer valence ratings of words from co-occurrence statistics is the one adopted by Louwerse and Recchia (2014), who followed a bootstrapping method to extend the ANEW lexicon (Bradley and Lang 1999). Another approach is to exploit a resource such as SenticNet (Cambria et al. 2016) to infer valence based on values of polarity for words or conceptual primitives. As shown in Bondielli, Passaro, and Lenci (2017), a third viable strategy is to infer word valence from an emotive resource such as ItEM (Passaro, Pollacci, and Lenci 2015; Passaro and Lenci 2016), a distributional lexicon for Italian, in which words are associated with an emotive score for 8 different emotions. This solution has several advantages. Firstly, ItEM is based on an unsupervised method to estimate affective scores, that guarantees high coverage over Italian words and can be easily expanded, allowing for a quick adaptation to different contexts. Moreover, associating words with fine-grained emotional values allows for a wide range of analyses, such as for instance hate and violence detection in texts.

The vectors used in Bondielli, Passaro, and Lenci (2017) relies on a classical count-based distributional model, and have provided interesting results. Based on these findings, in this work we focus on whether and how results could be improved by exploiting word embeddings learnt with a prediction-based model (Lenci 2018) to compute the affective scores. The proposed strategy is expected to perform better than the count-based one, and it is backed by an extensive body of related work in which Sentiment Lexicons are created by exploiting dense word vector representations obtained with neural network models (Bengio et al. 2003; Mikolov et al. 2013a, 2013b; Turian, Ratinov, and Bengio 2010; Huang et al. 2012). Moreover, such an approach has been successfully implemented in several Sentiment Analysis tasks (Tang et al. 2014; Yu et al. 2017; Castellucci, Croce, and Basili 2015, 2016; Basili, Croce, and Castellucci 2017).

This paper is organized as follows: In section 2 we describe the resources employed in this research, namely ANEW (section 2.1) and ItEM (section 2.2). In section 2.3, we present additional versions of ItEM based on neural embeddings. In section 3, we present three methods to infer valence ratings starting from distributional emotive scores: In the first two experiments, like in Bondielli, Passaro, and Lenci (2017), we predict a continuous valence score by exploiting a polynomial regression model (section 3.1) and a discrete score by means of logistic regression (section 3.2). In a third experiment (section 3.3), we present a new method that uses emotive seeds to predict a word valence. In this latter case, to assess the reliability of the method, we measure the correlation between the predicted scores and the human rated ones in ANEW. All experiments have been carried out with the count-based and the prediction-based versions of ItEM, to compare the effect of these two families of distributional models to learn the affect of lexical items. Finally, in section 4 we discuss our results and findings.

2. Affective resources

The main goal of this paper is to show that distributional emotive and affective scores can be used to infer a word's valence, as a crucial piece of information to determine the affective content of texts. Our research relies on two main resources, which we describe in this section: The Italian version of the Affective Norms for English Words (ANEW)

(Montefinese et al. 2014) and the Italian EMotive lexicon (ItEM) (Passaro, Pollacci, and Lenci 2015; Passaro and Lenci 2016).

2.1 Italian ANEW

ANEW (Bradley and Lang 1999) is a database containing 1034 English words rated for *valence*, *arousal* and *dominance*. The affective rating system used to annotate words is a variant of the Self-Assessment Manikin (SAM: Lang (1980)). The SAM is a technique built with the aim to assess the affective reaction of a person to different kinds of stimuli, in terms of pleasure, arousal and dominance (Bradley and Lang 1994). In ANEW, the SAM uses a numerical scale, ranging from 1 to 9, and is applied to all the main variables. For example, if we consider the valence, the rate 1 means unpleasant and 9 means very pleasant.

Connotation is a cultural phenomenon that may vary greatly between languages and different time spans (Das and Bandyopadhyay 2010) and, consequently, the "correct" translation of a word can have a different emotional connotation in different languages (Chen, Kennedy, and Zhou 2012). For this reason, the collection of affective norms has been carried out for many languages including Italian. The Italian adaptation of ANEW contains the norms for the translation of the original ANEW words, as well as for words taken from the Italian Semantic Norms (Montefinese et al. 2013). The total number of annotated words is 1,121. The three main dimensions of valence, arousal and dominance were rated using again the SAM scale, in order to provide consistency with the original norms. Apart from the original affective ratings, new dimensions were collected as well, namely subjective and objective psycholinguistic indexes. Subjective indexes are *familarity*, *imageability*, and *concreteness*. The familiarity index is based on subjective measures of how often participants both use and are exposed to a given word (Montefinese et al. 2014); Concreteness is the extent to which a word is tangible (Paivio, Yuille, and Madigan 1968); Imageability refers to the ease of generating a mental image for a word (Paivio, Yuille, and Madigan 1968). Objective indexes represent features of a word, such as length, frequency in two corpora (CoLFIS (Bertinetto et al. 2005) and La Repubblica (Baroni et al. 2004)), and number of orthographic neighbors. For the affective ratings, researchers also held into account gender differences. For example, a word like *allegro* "merry" has a very high rating for valence (8.11), and relatively high ratings for arousal and dominance (5.89 and 6.86 respectively). On the other end of the spectrum, *afflizione* "grief" is rated very low for valence (1.94), but it is considered a medium-high arousing word (6.39) and a medium-low dominance word (3.18).

The experiments were conducted on 1,084 participants, all native speakers and undergraduate psychology students. Out of all the participants, 684 were used to rate words with valence, arousal and dominance scores, and 400 to perform familiarity, imageability and concreteness evaluations. Each word was rated by at least 31 participants (of whom at least 10 male) for affective ratings, and by at least 20 participants for psycholinguistic ratings. Participants were asked to rate words using the SAM scale for affective ratings. The final resource is therefore composed of the original ANEW word, its Italian translation, and mean scores and standard deviation for each of the considered dimension. For affective ratings, measurement are also reported for male and female participants.

The main contribution of the Italian ANEW to the present research is that it provides us with an highly controlled scoring for affective ratings, that can be easily exploited to evaluate affective distributional scores.

2.2 ItEM

The Italian EMotive lexicon (ItEM) is a distributional resource described in Passaro, Pollacci, and Lenci (2015), Passaro and Lenci (2016), and based on the so-called Distributional Hypothesis (Harris 1954), which states that semantically similar words tend to appear in similar contexts. In ItEM, this hypothesis has been generalized to emotions, as follows:

A word w is associated with an emotion e if it co-occurs in similar contexts of other words associated with e.

To implement this hypothesis, in the basic version of ItEM, each emotion has been represented as a centroid vector built out of a set of seed words strongly associated to each of the target emotions.

The resource has been developed in a three stage process. The first phase was devoted to the collection a small set of seed words highly associated with one of Plutchik's basic emotions (Plutchik 1980): JOY, SADNESS, ANGER, FEAR, TRUST, DISGUST, SURPRISE and ANTICIPATION. In a second phase, distributional semantic methods were exploited to expand the seeds and populate the resource. Finally, the automatically extracted emotive annotations have been evaluated via crowdsourcing.

The goal of the first phase was to collect a small lexicon of *emotive lexemes*, highly associated to one or more Plutchik's basic emotions. Such a goal was reached by means of an online feature elicitation paradigm, in which 60 Italian native speakers were asked to list, for each emotion, 5 lexical items for each of our PoS of interest (Nouns, Adjectives and Verbs). After applying various filters and revisions, we obtained a lexicon of 347 words. For each word in this set, its emotion distinctiveness score was calculated – following Devlin et al. (1998) – as its informativeness (i.e., the reciprocal of the number of emotions for which the word was generated). For example, the distinctiveness of the word *amore* "love" is 1/3, given the following distribution of its production frequency: JOY = 2, TRUST = 5, and ANTICIPATION = 4. The seeds were restricted to the words with a distinctiveness score equal to 1 (i.e., the words produced/evoked by a single emotion). In addition, this set was expanded with the names of the emotions such as *gioia* "joy" , *rabbia* "anger" and their synonyms attested in Multiwordnet (Pianta, Bentivogli, and Girardi 2002) and Treccani Online Dictionary[1] for a total of 555 emotive seeds.

In the bootstrapping phase, a count-based Distributional Semantic Model (DSM) was used to expand the seeds using a corpus-based model inspired to Turney and Littmann (Turney and Littman 2003) to automatically infer the semantic orientation of a word from its distributional similarity with a set of positive and negative words. In particular, the DSM was created by extracting from La Repubblica corpus (Baroni et al. 2004) and itWaC (Baroni et al. 2009) the list of the 30,000 most frequent nouns, verbs and adjectives and recording their co-occurrences within a five word symmetric window centered on the target word. Co-occurrences were reweighted with Positive Pointwise Mutual Information (PPMI) (Church and Hanks 1990), but with negative values raised to 0. To optimize the vector space, we followed the approach in Polajnar and Clark (2014) and we selected the top 240 contexts for each target word. As a last step, we applied singular value decomposition (SVD), to reduce the matrix to 300 dimensions.

Adapting the approach Turney and Littman (2003), the emotions were represented as centroid vectors built from the mean of the vectors of the relative seeds. For each

1 http://www.treccani.it/vocabolario/.

emotion E, we computed a word emotive score σ by measuring the cosine similarity of the word vector $\overrightarrow{\boldsymbol{w}}$ in the DSM with the centroid vector of E ($\overrightarrow{\boldsymbol{C}_{\mathrm{E}}}$):

$$\sigma(E, w) = \frac{\overrightarrow{\boldsymbol{w}} \cdot \overrightarrow{\boldsymbol{C}_{\mathrm{E}}}}{||\overrightarrow{\boldsymbol{w}}|| \cdot ||\overrightarrow{\boldsymbol{C}_{\mathrm{E}}}||} \tag{1}$$

This score measures the association of a word with a given emotion. For instance, the amount of ANGER associated with the noun *gelosia* "jealousy" is estimated with the cosine similarity between the vector of *gelosia* and the centroid vector of ANGER. The following is the emotion distribution of that word, modeled with the cosine similarity with the emotive centroids: ANGER: 0.65; DISGUST: 0.43; FEAR: 0.36; SADNESS: 0.32; JOY: 0.24; SURPRISE: 0.24; ANTICIPATION : 0.20; TRUST : 0.12.

ItEM was evaluated with two crowdsourcing tasks on the Crowdflower (CF) platform[2] to compare the model performance on a random set of words, including also possibly neutral words, associated with human ratings about their association or lack of association with emotions. The details and results of the ItEM evaluation are reported in Passaro and Lenci (2016).

2.3 Adapting ItEM to prediction-based word embeddings

In order to adapt ItEM to prediction-based word embeddings, we developed a new model, namely the ITEM-8-PREDICT, in which the vectors of the words were built with the state-of-the-art prediction-based DSM Word2vec (Mikolov et al. 2013a, 2013b). In particular, the neural word embeddings were trained on the lemmatized concatenation of the corpora La Repubblica (Baroni et al. 2004) and itWaC (Baroni et al. 2009), by restricting the vocabulary to nouns verbs and adjectives and representing each token in the form $< lemma - PoS >$. After testing few configurations, we used the Skip-Gram with Negative Sampling algorithm with the following hyperparameters: the size of the embedding was set to 500 for each word; the context span was set to 5; the occurrence threshold was set to $1 * e^{-4}$, and the number of negative examples was set to 10. For the sake of comparison, we decided to implement a 500 dimensions vector for the count model as well, which will be referred as ITEM-8-COUNT for the rest of the paper.

3. From fine-grained Emotion Values to Polarity

To predict valence ratings from the distributional emotive scores, we performed several experiments. In Bondielli, Passaro, and Lenci (2017), we showed two alternative methods to predict, respectively, a continuous and a discrete valence rating by exploiting distributional emotive scores. In particular, we used a polynomial and a logistic regression model to infer valence from emotions. In this work we explore this problem more deeply, and propose new distributional methods to construct valence lexicons.

For the sake of comparison, we conducted our experiments on the same dataset analyzed in Bondielli, Passaro, and Lenci (2017). First of all, a simple preprocessing phase was applied to align Italian ANEW and ItEM. The former includes 1,121 words, but 65 of them have multiple PoS (e.g., *aereo* "plane" can be both a noun and an adjective). We

2 http://www.crowdflower.com.

duplicated each word, extending the dataset to 1,189 elements, and extracted distinct emotive scores for each <lemma,PoS> pair. In addition, we replaced word forms like *scorie* "waste", with their most frequent word type (*scoria*) in ItWaC and La Repubblica.

Eventually, in all the experiments, some ANEW words were left out of the analysis because they were not covered by the current version of ItEM. This happened for two reasons. Firstly, the word was not included among the ItEM target terms (i.e., low-frequency words not appearing in the list of the top 30,000 words of the considered corpora). Secondly, the words had a negative cosine values with all the emotive centroids in ItEM. In each experiment, we report the coverage with respect to Italian ANEW.

3.1 Predicting a continuous valence score with polynomial regression

As shown in Figure 1, the distribution of Italian ANEW data is bimodal and therefore, we used a polynomial regression model to predict the valence of words in ANEW with their emotive scores in ItEM.

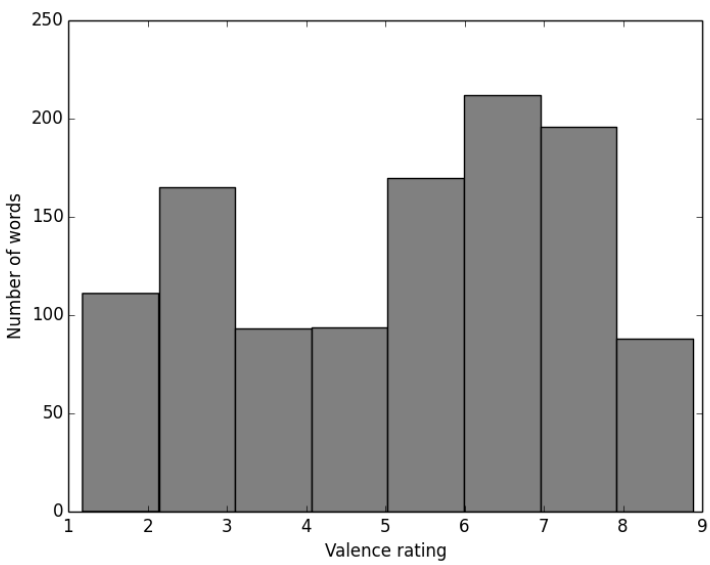


Figure 1
Distribution of valence ratings for Italian ANEW Italian. The histogram clearly shows a lower number of examples for valence ratings in range $[3, 5]$ and for very high and very low values. On the contrary, words with a valence rating in the ranges $[5, 8]$ and $[2, 3]$ are well represented, with a slight bias towards higher values.

To define the most performing degree (Deg) of the polynomial function, we carried out 10-fold cross validation for degrees in the range $[1, 5]$. We can clearly identify overfitting starting from degrees equal or higher than 3 (cf. Table 1). This is due to the fact that, given the number of parameters ($\#P$) for regression, we can estimate the minimum number of observations (Min. Obs.) needed to avoid overfitting. This number was computed as $\#P \times 15$, and should be smaller or equal to the total number of observations used to build the model. In our case, this was true only for polynomial of degree 1 and 2. This finding is in line with Schmidt (1971) and Harrell (2001). In their

work, they demonstrated that to guarantee the reliability of the prediction, for each parameter in the regression model there should be a minimum number of observations between 10 an 20 in the data.

Table 1
Experiments performed to define the best degree (Deg) for the polynomial. For polynomial of degree 1 and 2 we can see an increase in the computed R^2. For higher degrees, the minimum number of observation exceeds the size of our dataset. This causes the MSE to decrease, but R^2 drastically drops as well. The best performing degree for our dataset with respect to R^2 and MSE is degree 2.

Deg	#P	Min. Obs.	R^2	MSE
1	9	~ 135	0.46	2.24
2	45	~ 675	0.53	1.82
3	165	~ 2475	0.31	1.50
4	495	~ 7425	-81.29	0.96
5	1287	~ 19305	-11 B	0.00

Given these results, we decided to use a degree 2 for the interpolation of our parameters. We built two models and compared their results. The first model, which we called COUNT, replicates the model presented in Bondielli, Passaro, and Lenci (2017), and exploits the emotive scores in ITEM-8-COUNT, the only difference being vector dimensionality, which was now set to 500. The second model, which we called PREDICT, was built by exploiting the emotive scores in ITEM-8-PREDICT.

We performed polynomial interpolation of the parameters (i.e., the distributional emotive values), and applied a simple multiple linear regression over the new data in order to predict valence. Results of this experiment are shown in Table 2. First of all, we show how the models predict the actual ANEW valence ratings by exploiting the whole dataset. Then, we perform 10-fold cross validation in order to better assess the predictive capabilities of our DSMs. The results, where R-squared (R^2), mean absolute error (MeanAE), mean squared error (MSE), and median absolute error (MedianAE) were used for evaluation, are shown in Table 3.

Table 2
Results of the evaluations. Both models are based on the analysis of 1,090 data points, i.e. the words contained in both ItEM and ANEW. Prediction-based word embeddings show improvements for predicting the whole dataset and for 10-fold cross validation (CV). More specifically, R^2 is increased by 5 points, and all the mean and median errors are reduced.

Model	R^2	*MeanAE*	*MSE*	*MedianAE*
COUNT	0.64	0.98	1.54	0.81
COUNT - CV	0.61	1.01	1.65	1.01
PREDICT	0.69	0.89	1.29	0.72
PREDICT - CV	0.66	0.93	1.41	0.93

The results show the same trend for both the COUNT and PREDICT model. We see that the difference between human-rated valence and predicted valence is on average around 1 (it falls between 0.9 and 1.5). However, the results also show that the PREDICT model clearly outperforms the COUNT model for what concerns R^2. This means that a

distributional emotive space such as ItEM can benefit from using prediction-based word embeddings to compute the emotive connotation of words. In addition, the Pearson's correlation coefficient between predicted and human-rated valences is increased from 0.8 ($p < 0.005$) of the COUNT model to 0.83 $p < 0.005$ of the PREDICT one. In both cases, correlation is very high, proving the excellent ability of DSMs to model behavioral data about word affective valence.

It is also important to stress that the use of word embeddings improved the results for MeanAE and MSE. This means that the predictions of the PREDICT model are on average closer to the actual valence ratings of words. This is crucial in order to improve performances for words with medium valence ratings. In fact, the model presented in (Bondielli, Passaro, and Lenci 2017) performed better on low-valenced or high-valenced words. Medium-valenced words on the contrary had more chances to be predicted as either too high or too low, given the mean errors of the model. The PREDICT model, albeit not perfect, may be less prone to this kind of problem, given a generally smaller average error.

3.2 Predicting a discrete valence score with Logistic regression

Following the approach presented in Bondielli, Passaro, and Lenci (2017), we performed a second experiment to evaluate the results of a logistic regression classifier aimed at predicting a discrete valence score. The discretization of the *gold* valence was performed by considering as POSITIVE the words with $valence >= 5.5$, and as NEGATIVE the others. Again, we compared the two versions of ItEM, that is ITEM-8-COUNT and ITEM-8-PREDICT. The goal was to predict a *binary valence* and assess the differences between count- and prediction-based DSMs. Results of these experiments are shown in Table 3.

Table 3
Logistic regression (Cross Validation). Both models are based on the analysis of 1,090 data points, i.e. the words contained in both ItEM and ANEW. Precision, Recall and F1 are improved by exploiting prediction-based embeddings to build ItEM

Model	*Precision*	*Recall*	*F1*
COUNT - MACROAVG	0.828	0.820	0.821
PREDICT - MACROAVG	0.844	0.841	0.842

The results of this experiment again show how the use of neural embeddings can improve the classification performances. The COUNT model has an average F1 of 0.82, whereas the PREDICT model scores 0.84 on the same data.

3.3 Predicting a polarity score with a valence version of ItEM

In a third experiment, we created a distributional polarity lexicon in which Italian words were associated with a positiveness and a negativeness score, rather than the 8 emotive scores described in section 2.2.

First of all, we splitted the emotions into a positive and a negative group. In particular, the seeds elicitated for the emotions JOY and TRUST have been grouped into the class POSITIVE and the seeds elicitated for SADNESS, ANGER, FEAR and DISGUST have been classified as NEGATIVE. The emotions SURPRISE and ANTICIPATION have

been left out of the analysis because of their mixed nature. More specifically we selected the words with a cue validity higher than 0.6 for the original emotion and a production frequency higher than 1. Globally, we selected 119 positive seeds and 310 negative ones. The bootstrapping phase was performed as with the ITEM-8 models. This way, we built two new models, namely ITEM-2-COUNT trained with a count-based DSM (cf. section 2.2), and ITEM-2-PREDICT trained with Word2vec (cf. section 2.3).

To evaluate this method, we approximated the polarity of a word w with the difference between its positiveness (eq. 3) and negativeness (eq. 4) score:

$$polarity(w) = positiveness(w) - negativeness(w) \tag{2}$$

Both scores were calculated as the cosine similarity between the vector of the word ($\overrightarrow{\boldsymbol{w}}$) and the centroid vector of positiveness ($\overrightarrow{\boldsymbol{C}_{\mathrm{P}}}$) and negativeness ($\overrightarrow{\boldsymbol{C}_{\mathrm{N}}}$):

$$positiveness(w) = \frac{\overrightarrow{\boldsymbol{w}} \cdot \overrightarrow{\boldsymbol{C}_{\mathrm{P}}}}{||\overrightarrow{\boldsymbol{w}}|| \cdot ||\overrightarrow{\boldsymbol{C}_{\mathrm{P}}}||} \tag{3}$$

$$negativeness(w) = \frac{\overrightarrow{\boldsymbol{w}} \cdot \overrightarrow{\boldsymbol{C}_{\mathrm{N}}}}{||\overrightarrow{\boldsymbol{w}}|| \cdot ||\overrightarrow{\boldsymbol{C}_{\mathrm{N}}}||} \tag{4}$$

A polarity score close to 1 indicates positiveness while a score close to -1 means negativeness.

In these experiments, we measured the correlation coefficient between Valence and Polarity. Table 4 shows the results of the correlation between the valence in ANEW and the polarity calculated using count-based vs. prediction-based semantic vectors.

Table 4
Correlation coefficient between the Valence in ANEW and the Polarity produced using ITEM-2-COUNT and ITEM-2-PREDICT. We provide both Pearson and Spearman correlation coefficients. In all the experiments we found a $p - value < 0.001$. Both models are based on the analysis of 1,090 data points, i.e. the words contained in both ItEM and ANEW.

Model	*Pearson r*	*Spearman ρ*
COUNT	0.743	0.777
PREDICT	0.785	0.794

The results of this experiment show that the distributional polarity highly correlates with human-elicited data. Moreover, once again the use of word embeddings improves the prediction. These results, compared with the ones obtained with the polynomial regression model (see section 3.1), prove that this method is a reliable alternative to predict valence from polarity, but, at the same time, that a more granular emotion taxonomy, when available, is the best option.

Moreover, by discretizing both valence and polarity with the thresholds used in section 3.2, we observe that the binary models, especially count ones, despite achieving

acceptable accuracy, are outperformed by distributional models relying on a richer emotion taxonomy. The results of this experiment are shown in Table 5.

Table 5
Performance of the discretized model. The discretization of the *gold* valence was performed by considering as POSITIVE the words with $valence >= 5.5$, and as NEGATIVE the others. The polarity was discretized by considering its sign (i.e., the words with a Polarity higher than 0.0 were considered as POSITIVE).

Model	*Precision*	*Recall*	*F1*
COUNT - MACROAVG	0.796	0.791	0.785
PREDICT - MACROAVG	0.827	0.828	0.826

4. Discussion

Overall, our experiments demonstrated that the valence of words can be inferred by means of both emotions and polarity estimated with Distributional Semantic Models. In particular, we showed three methods to infer valence ratings starting from distributional emotive scores: in the first two experiments, inspired by Bondielli, Passaro, and Lenci (2017), we predicted a continuous valence score by exploiting a polynomial regression model (section 3.1) and a discrete score by means of logistic regression (section 3.2). In a third experiment (section 3.3), we showed a method to infer valence directly by exploiting emotive seeds.

All experiments have been carried out with the count-based and the prediction-based versions of ItEM, to compare the effect of these two families of distributional models. In the first experiment we found that the use of word embeddings improves the performance despite the presence of medium valence words, which are supposedly the most difficult to classify. In the second one, we showed that by discretizing the valence into two polarity classes, such an improvement becomes more pronounced, by reaching an F1 of 0.84. Also in this case the model benefits from the use of prediction-based word vectors. Finally, in the third experiment we directly exploited a binary categorization of emotions to infer the valence and we found a high correlation between predicted and human rated valence. However, the model produced by this experiment, albeit being able to reliably predict valence from polarity, suffers from worse performances with respect to the model presented in the first experiment. This demonstrates that a more granular emotion taxonomy might be a better option. Due to the superiority of the PREDICT model in all the experiments we performed, we decided to deeper investigate the differences in terms of correlation between the two distributional methods aimed at predicting a continuous value of valence (experiment 1 and experiment 3). In particular, we studied the effect of the frequency and of the part of speech on the performances of the two models.

For what concerns frequency, we divided the dataset into three equally sized frequency classes. In this case, all the experiments showed the absence of a statistically significant difference between the COUNT and the PREDICT model for low frequency words. In all the other cases, the PREDICT model seems to work significantly better than the COUNT one.

As for Part of Speech, the results are shown in detail in Table 6. Although overall the difference between the PREDICT and the COUNT model is statistically significant

in terms of correlation between the ANEW valence and the distributional polarity, we found such difference to be not significant in the case of verbs.

Table 6
Performance in terms of Pearson's correlation divided by Part of Speech. The table reports the model and the PoS (together with the number of items in the test set).

Model	*Overall [1090]*	*Nouns [782]*	*Verbs [51]*	*Adj [254]*
COUNT (8 EMOTIONS)	0,80	0,79	0,78	0,85
PREDICT (8 EMOTIONS)	0,83	0,83	0,75	0,87
COUNT (2 EMOTIONS)	0,74	0,75	0,71	0,79
PREDICT (2 EMOTIONS)	0,79	0,78	0,71	0,82

Moreover, looking at verbs, we noticed a more pronounced drop in the correlation between actual and predicted values for the PREDICT model with respect to the COUNT one. In other words, the Δ between the correlation of verbs and overall results varies in the range 0.02 and 0.03 for the COUNT model while varying from 0.08 and 0.09 in the PREDICT one. It is clear that the dimension of the sample of the verbs affects the results (especially in the ITEM-8 experiment, in which the points in ANEW are directly embedded in the regression model), but these results open new questions about the behaviour of the prediction-based vectors to model the affective dimension of verbs. This suggests the existence of interesting differences between the two families of DSMs with respect to different PoS, a point we leave for future investigations.

5. Conclusions and ongoing research

In this work we studied the relationship between *valence* and distributional emotive scores inferred from count-based and prediction-based dense semantic vectors. We modeled our data with regression and correlation in order to predict both a continuous score for valence and its corresponding binomial version (i.e., polarity). The results we obtained in our experiments show both pros and cons of each approach. The exploitation of distributional emotive scores for predicting the valence rating for a word may prove advantageous because such scores can be easily obtained in an unsupervised way. Our experiments have in fact shown that, despite using relatively simple models such as polynomial and logistic regression and the creation of a polarity lexicon, we are able to infer valence ratings with good accuracy.

The experiments support two important conclusions:

- prediction-based DSMs produce significantly better lexical representations than count-based ones. This fact was already shown in number of semantic tasks by Baroni, Dinu, and Kruszewski (2014) and Mandera, Keuleers, and Brysbaert (2017). Our research is the first one to prove that this is true also to estimate the affective content of lexical items. Neural embeddings provide on average 3 points of improvement if we consider the Pearson's correlation and of 3 percentage points if we consider the F1 in the prediction of discrete valence;
- most research on Affective Computing focuses on valence defined as a binary category, but it is preferable to rely on a more granular emotion taxonomy, such as the one used by ItEM. Word valence can be better predicted by DSMs trained on 8 basic emotions, rather than DSMs directly trained on seeds grouped into a

positive and a negative class. Of course, this may also depend on the grouping criteria and on the fact that the seeds were originally collected with respect to their association with emotions rather than for their valence. We leave this point to further research.

One of the main drawbacks of our evaluation derives from the dimension of the ANEW dataset, and in particular from the lack of examples around the medium valence score ratings. It is clear that the ratings distribution in this resource prevented us from obtaining reliable results for continuous values. We are still confident that having access to a new resource covering the full spectrum of the valence more evenly would have a positive impact on our model. Despite the difficulties of modeling an accurate representation of a continuous valence rating from a small and unbalanced dataset like the Italian ANEW, we can identify a clear relationship between distributional emotional scores and a discrete valence obtained by categorizing the ratings into a positive and a negative class.

In the near future, we plan to improve the seeds used to build our distributional resources and to extend this work to predict sentiment polarity scores taken from SentiWordNet (Esuli and Sebastiani 2006a, 2006b), thereby exploiting the larger coverage of this resource. Moreover, we plan to follow the approach employed in ItEM to create a polarity lexicon for Italian, using ANEW words as seed to build positive and negative polarity centroids. In this case, we intend to evaluate the new resource with crowdsourcing or controlled psycholinguistic experiments. Finally, we aim at testing the effectiveness of our system for Sentiment Polarity Classification of texts.

References

Baroni, Marco, Silvia Bernardini, Federica Comastri, Lorenzo Piccioni, Alessandra Volpi, Guy Aston, and Marco Mazzoleni. 2004. Introducing the la repubblica corpus: A large, annotated, tei(xml)-compliant corpus of newspaper italian. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC 2004)*, pages 1171–1174, Lisbon, Portugal. European Language Resource Association (ELRA).

Baroni, Marco, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The wacky wide web: A collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation*, 43(3):209–226.

Baroni, Marco, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014)*, volume 1, pages 238–247, Baltimore, Maryland.

Basili, Roberto, Danilo Croce, and Giuseppe Castellucci. 2017. Dynamic polarity lexicon acquisition for advanced social media analytics. *International Journal of Engineering Business Management*, 9:1–18.

Bengio, Yoshua, Régean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155.

Bertinetto, Pier Marco, Cristina Burani, Alessandro Laudanna, Lucia Marconi, Daniela Ratti, Claudia Rolando, and Anna Maria Thornton. 2005. Corpus e lessico di frequenza dell'italiano scritto (CoLFIS). Technical report.

Bondielli, Alessandro, Lucia C. Passaro, and Alessandro Lenci. 2017. Emo2val: Inferring valence scores from fine-grained emotion values. In *Fourth Italian Conference on Computational Linguistics (CLiC-it 2017)*, pages 48–52, Rome, Italy. Accademia University Press.

Bradley, Margaret M. and Peter J. Lang. 1994. Measuring emotion: the self-assessment manikin and the semantic differential. *Journal of behavior therapy and experimental psychiatry*, 25(1):49–59.

Bradley, Margaret M. and Peter J. Lang. 1999. Affective norms for english words (ANEW): Instruction manual and affective ratings. Technical report, Technical report C-1, the center for research in psychophysiology, University of Florida.

Cambria, Erik, Soujanya Poria, Rajiv Bajpai, and Björn W. Schuller. 2016. Senticnet 4: A semantic resource for sentiment analysis based on conceptual primitives. In *Proceedings of the 26th*

International Conference on Computational Linguistics: Technical Papers (COLING 2016), pages 2666–2677, Osaka, Japan.
Castellucci, Giuseppe, Danilo Croce, and Roberto Basili. 2015. Acquiring a large scale polarity lexicon through unsupervised distributional methods. In Chris Biemann, Siegfried Handschuh, André Freitas, Farid Meziane, and Elisabeth Métais, editors, *Proceedings of the 20th International Conference on Applications of Natural Language to Information Systems (NLDB 2015)*, pages 73–86, Passau, Germany. Springer International Publishing.
Castellucci, Giuseppe, Danilo Croce, and Roberto Basili. 2016. A language independent method for generating large scale polarity lexicons. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 38–45, Portorož, Slovenia.
Chen, Stephen H., Morgan Kennedy, and Qing Zhou. 2012. Parents' expression and discussion of emotion in the multilingual family: Does language matter? *Perspectives on Psychological Science*, 7(4):365–383.
Church, Kenneth W. and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16:22–29.
Das, Amitava and Sivaji Bandyopadhyay. 2010. Towards the global SentiWordNet. In *Proceedings of the 24th Pacific Asia Conference on Language, Information and Computation (PACLIC 2010)*, pages 799–808, Sendai, Japan.
Devlin, Joseph T., Laura M. Gonnerman, Elaine S. Andersen, and Mark S. Seidenberg. 1998. Category-specific semantic deficits in focal and widespread brain damage: A computational account. *Journal of cognitive Neuroscience*, 10(1):77–94.
Esuli, Andrea and Fabrizio Sebastiani. 2006a. Determining term subjectivity and term orientation for opinion mining. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL'06)*, pages 193–200, Trento, Italy. Association for Computational Linguistics.
Esuli, Andrea and Fabrizio Sebastiani. 2006b. SentiWordNet: A publicly available lexical resource for opinion mining. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006)*, pages 417–422, Genoa, Italy. European Language Resource Association (ELRA).
Harrell, Frank E. 2001. *Regression Modeling Strategies with Applications to Linear Models, Logistic Regression, and Survival Analysis*. Springer New York.
Harris, Zelig. 1954. Distributional structure. *Word*, 10(23):146–162.
Huang, Eric H., Richard Socher, Christopher D. Manning, and Andrew Y. Ng. 2012. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers (ACL 2012)*, volume 1, pages 873–882, Jeju Island, Korea. Association for Computational Linguistics.
Lang, Peter J. 1980. Behavioral treatment and bio-behavioral assessment: Computer applications. In Joseph B. Sidowski, James H. Johnson, and Thomas A. Williams, editors, *Technology in Mental Health Care Delivery Systems*. Ablex Pub. Corp., Norwood, NJ, pages 119–137.
Lang, Peter J., Margaret M. Bradley, and Bruce N. Cuthbert. 1997. International affective picture system (IAPS): Technical manual and affective ratings. *NIMH Center for the Study of Emotion and Attention*, pages 39–58.
Lenci, Alessandro. 2018. Distributional models of word meaning. *Annual Review of Linguistics*, 4:151–171.
Louwerse, Max M. and Gabriel Recchia. 2014. Reproducing affective norms with lexical co-occurrence statistics: Predicting valence, arousal, and dominance. *The Quarterly Journal of Experimental Psychology*, 68(12):1–15.
Mandera, Paweł, Emmanuel Keuleers, and Marc Brysbaert. 2017. Explaining human performance in psycholinguistic tasks with models of semantic similarity based on prediction and counting: A review and empirical validation. *Journal of Memory and Language*, 92:57–78.
Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *Computing Research Repository (CoRR)*.
Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems (NIPS 2013)*, volume 2, pages 3111–3119, Lake Tahoe, Nevada. Curran Associates Inc.
Montefinese, Maria, Ettore Ambrosini, Beth Fairfield, and Nicola Mammarella. 2013. Semantic memory: A feature-based analysis and new norms for italian. *Behavior Research Methods*, 45(2):440–461.

Montefinese, Maria, Ettore Ambrosini, Beth Fairfield, and Nicola Mammarella. 2014. The adaptation of the affective norms for english words (ANEW) for italian. *Behavior Research Methods*, 46(3):887–903.

Paivio, Allan, John C. Yuille, and Stephen A. Madigan. 1968. Concreteness, imagery, and meaningfulness values for 925 nouns. *Journal of experimental psychology*, 76(1p2):1–25.

Passaro, Lucia C. and Alessandro Lenci. 2016. Evaluating context selection strategies to build emotive vector space models. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Portorož, Slovenia.

Passaro, Lucia C., Laura Pollacci, and Alessandro Lenci. 2015. Item: A vector space model to bootstrap an italian emotive lexicon. In *Proceedings of the Second Italian Conference on Computational Linguistics (CLiC-it 2015)*, pages 215–220, Trento, Italy. Academia University Press.

Pianta, Emanuele, Luisa Bentivogli, and Christian Girardi. 2002. Multiwordnet: developing an aligned multilingual database. In *Proceedings of the First International Conference on Global WordNet (GWC2002)*, pages 293–302, Mysore, India.

Plutchik, Robert. 1980. General psychoevolutionary theory of emotion. *Emotion: Theory, Research, and Experience*, 1:3–33.

Polajnar, Tamara and Stephen Clark. 2014. Improving distributional semantic vectors through context selection and normalisation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, (EACL 2014)*, pages 230–238, Gothenburg, Sweden. Association for Computational Linguistics.

Schmidt, Frank L. 1971. The relative efficiency of regression and simple unit predictor weights in applied differential psychology. *Educational and Psychological Measurement*, 31(3):699–714.

Tang, Duyu, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. 2014. Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014)*, volume 1: Long Papers, pages 1555–1565, Baltimore, Maryland, June. Association for Computational Linguistics.

Turian, Joseph, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010)*, pages 384–394, Uppsala, Sweden. Association for Computational Linguistics.

Turney, Peter D. and Michael L. Littman. 2003. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Management Information Systems (TMIS)*, 21(4):315–346.

Warriner, Amy Beth, Victor Kuperman, and Marc Brysbaert. 2013. Norms of valence, arousal, and dominance for 13,915 english lemmas. *Behavior research methods*, 45(4):1191–1207.

Yu, Liang-Chih, Jin Wang, K Robert Lai, and Xuejie Zhang. 2017. Refining word embeddings for sentiment analysis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*, pages 534–539, Copenhagen, Denmark. Association for Computational Linguistics.

Bi-directional LSTM-CNNs-CRF for Italian Sequence Labeling and Multi-Task Learning

Pierpaolo Basile*
Università degli Studi di Bari Aldo Moro

Pierluigi Cassotti**
Università degli Studi di Bari Aldo Moro

Lucia Siciliani†
Università degli Studi di Bari Aldo Moro

Giovanni Semeraro‡
Università degli Studi di Bari Aldo Moro

In this paper, we propose a Deep Learning architecture for several Italian Natural Language Processing tasks 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. This architecture provided state of the art performance in several sequence labeling tasks for the English language. We exploit the same approach for the Italian language and extend it for performing a multi-task learning involving PoS-tagging and sentiment analysis. 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 the Italian.

1. Background and Motivation

Deep Learning (DL) gained a lot of attention in last years for its capacity to generalize models without the need of feature engineering and its ability to provide good performance. On the other hand, good performance can be achieved by accurately designing the architecture used to perform the learning task. In Natural Language Processing (NLP) several DL architectures have been proposed to solve many tasks, ranging from speech recognition to parsing. Some typical NLP tasks, such as part-of-speech (PoS) tagging and Named Entity Recognition (NER), can be solved as sequence labeling problem. Traditional high performance NLP methods for sequence labeling are linear statistical models, including Conditional Random Fields (CRF) and Hidden Markov Models (HMM) (Ratinov and Roth 2009; Passos, Kumar, and McCallum 2014; Luo et al. 2015), which rely on hand-crafted features and task/language specific resources. However, developing such task/language specific resources has a cost. Moreover, it makes difficult to adapt the model to new tasks, new domains or new languages.

In (Ma and Hovy 2016), the authors propose a state of the art sequence labeling method based on a neural network architecture that benefits from both word- and character-level representations through the combination of bidirectional LSTM, CNN

* Department of Computer Science, Via E. Orabona, 4 - 70125 Bari (Italy).
E-mail: pierpaolo.basile@uniba.it
** Department of Computer Science, Via E. Orabona, 4 - 70125 Bari (Italy).
E-mail: pierluigicassotti@gmail.com
† Department of Computer Science, Via E. Orabona, 4 - 70125 Bari (Italy).
E-mail: lucia.siciliani@uniba.it
‡ Department of Computer Science, Via E. Orabona, 4 - 70125 Bari (Italy).
E-mail: giovanni.semeraro@uniba.it

and CRF. The method is able to achieve state of the art performance in sequence labeling tasks for the English with no need of using hand-crafted features.

In this paper, we exploit the aforementioned architecture for solving three NLP tasks in Italian: PoS-tagging of tweets, NER and Super Sense Tagging (SST). We have already proposed an evaluation on these tasks (Basile, Semeraro, and Cassotti 2017) using the same architecture, but without correctly optimizing hyperparameters due to the lack of a validation set. In this paper, we describe a procedure for hyperparameters optimization based on k-fold cross-validation. Moreover, we extend this architecture for performing a multi-task learning (Zhang and Yang 2017; Ruder 2017) involving PoS-tagging and sentiment analysis. This has been possible by using training data with multiple levels of annotations. In particular, we exploit training data about tweets annotated with PoS-tag, polarity and irony.

Our research goal is twofold: 1) to prove the effectiveness of the DL architecture in a different language - in this case Italian - without using language specific features; 2) to investigate the performance of the architecture in the context of multi-task learning, when multiple levels of annotations on the same training set are exploited.

The results of the evaluation prove that our approach is able to achieve state of the art performance and in some cases it is able to overcome the best systems developed for the Italian using no specific language resources.

The paper is structured as follows: Section 2 provides details about our methodology and summarizes the DL architecture proposed in (Ma and Hovy 2016), while Section 3 shows the results of the evaluation. Final remarks are reported in Section 4.

2. Methodology

Our approach relies on the DL architecture proposed in (Ma and Hovy 2016), where the authors combine two aspects previously exploited separately: 1) the use of a character-level representation (Chiu and Nichols 2015); 2) the addition of an output layer based on CRF (Huang, Xu, and Yu 2015). The architecture is sketched in Figure 1: The input level of the Convolutional Neural Network (CNN) is represented by the character-level representation. A dropout layer (Srivastava et al. 2014) with convolution and max pooling is applied before feeding the CNN with character embeddings. Then, the character embeddings are concatenated with the word embeddings to form the input for the Bi-directional LSTM (bi-LSTM) layer as sketched in Figure 2. The dropout layer is also applied to output vectors from the LSTM layer. The output layer is based on Conditional Random Fields (CRF) and it modifies the output vectors of the LSTM in order to find the best output sequence. The CRF layer is useful for learning correlations between labels in neighborhoods; for example, usually a noun follows an article in PoS-tagging, or the I-ORG tag[1] cannot follow the I-PER tag in the NER task.

The aforementioned architecture can be easily adapted to other languages since it does not rely on language dependent features. The only components outside the architecture are the word embeddings that can be built by relying on a corpus of documents of the specific language. In Section 3, we provide details about the setup of the architecture parameters and the building of word embeddings for Italian. In particular, we adopt two different word embeddings: One for PoS-tagging and one

1 The IOB2 schema for data annotation is usually adopted in the NER task.

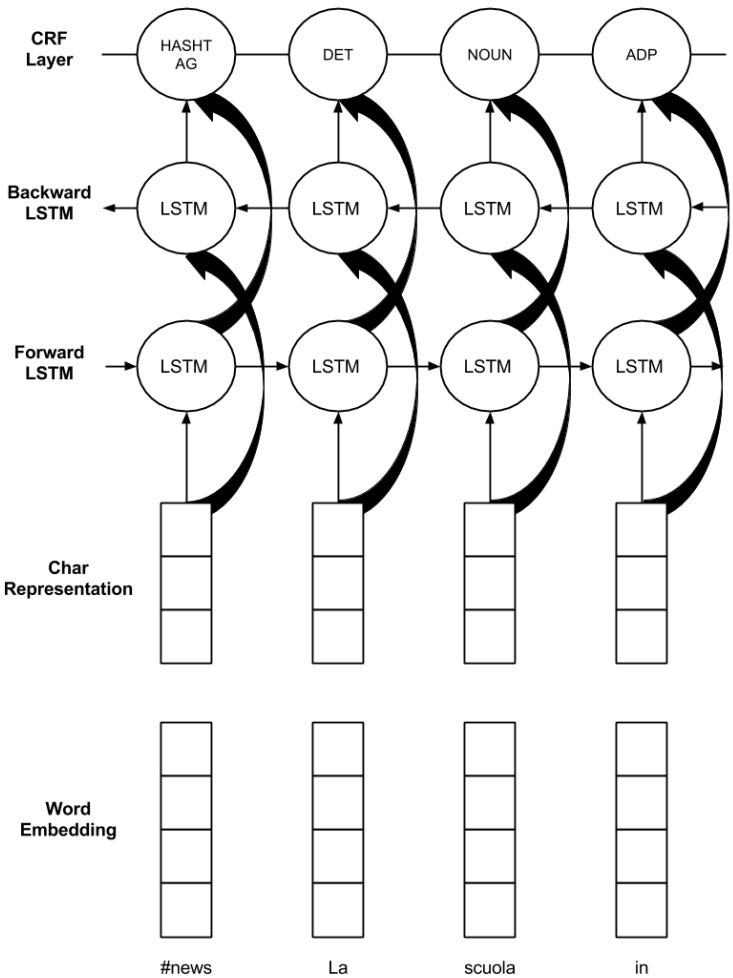


Figure 1
The DL architecture for sequence labeling.

for NER and SST. Moreover, we re-implemented[2] the architecture by using the Keras[3] framework and Tensorflow[4] as back-end.

2.1 Multi-task learning

We extend the previous architecture for performing multi-task learning. In particular, we want to jointly learn PoS-tag, polarity and irony. It is important to underline that PoS-tag is assigned to each token occurring in the sentence, while polarity and irony are

2 The code is available on line: `https://github.com/pippokill/bilstm-cnn-crf-seq-ita`
3 `https://keras.io/`
4 `https://www.tensorflow.org/`

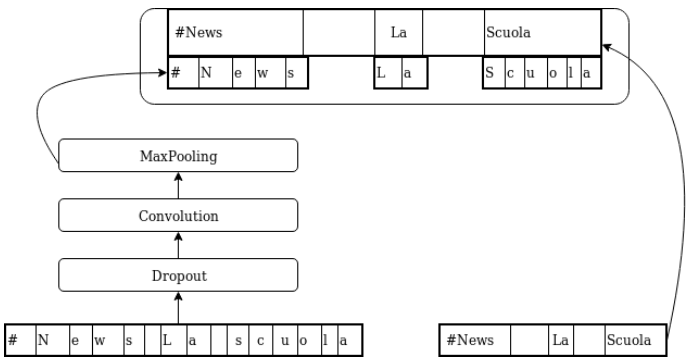


Figure 2
The input level of the DL architecture.

assigned to the whole sentence. We follow a hard parameter sharing approach (Ruder 2017) in which we have some shared layers in the bottom of the network and task-specific layers on the top.

The proposed architecture depends on the particular sentiment analysis task (Barbieri et al. 2016) that we want to perform. The task is deeply described in Section 3.4. Here, we want to point out that we need to solve four binary classification tasks: Subjectivity (true/false), positive polarity (true/false), negative polarity (true/false) and irony (true/false). We want to train a classifier for these classes jointly with the PoS-tagging task.

For this purpose, we add a parallel layer to the CRF one. In particular, a new layer based on a bi-LSTM is added using the same dimension of the first LSTM layer. Then, a dropout layer is applied and the final classes probabilities are computed by a binary cross entropy function for each class. In this case the last layer does not predict a tag for each token, but it predicts only one tag[5] for each classification task (subjectivity, positive, negative, irony). The multi-task architecture is sketched in Figure 3.

We can notice that the output of the first bi-LSTM layer is the input of the CRF layer for predicting PoS-tags, while each binary sentiment task is implemented by a new bi-LSTM layer and a cross entropy function.

3. Evaluation

We provide an evaluation in the context of four tasks for the Italian language. The first three tasks concern sequence labeling: 1) PoS-tagging of Italian tweets; 2) NER of Italian news 3) Super Sense Tagging. The fourth task concerns sentiment classification on Twitter. In such task, we try to jointly learn PoS-tagging and sentiment classification.

All tasks are performed using Italian datasets. In particular we exploit data coming from the last edition (2016) of EVALITA[6] (Basile et al. 2016) and the previous ones (2009 (Magnini and Cappelli 2009) and 2011[7]). EVALITA[8] is a periodic evaluation campaign of NLP and speech tools for the Italian language. The usage of a standard benchmark

5 Assigned to the whole text.
6 https://github.com/evalita2016/data
7 http://www.evalita.it/2011/working_notes
8 http://www.evalita.it/

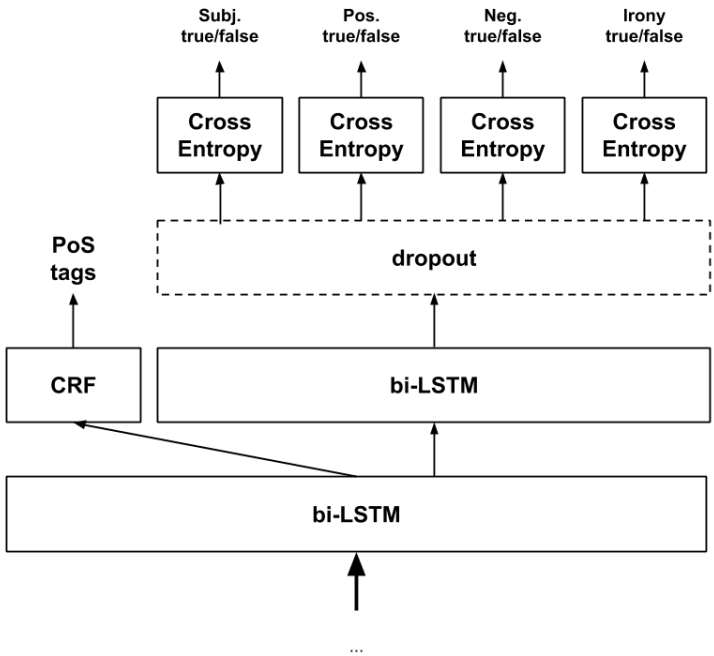


Figure 3
The DL architecture for multi-task learning.

allows us to compare our system with the state of the art approaches for the Italian language. In particular, EVALITA 2016 contains some shared-task data. We exploit these data for performing the multi-task evaluation.

Each task has its specific parameters, with parameters in Table 1 that are shared by all tasks.

Table 1
Parameters' values.

Parameter	Value
Framework	Keras 2.0.1
Back-end	Tensorflow 1.1.0
Char embed. dimension	30
Word embed. dimension	300
Window size	3
LTSM dimension	200 (bi-LTSM 400)
Gradient clipping	5.0
Dropout	0.5

We perform parameters optimization using 5-fold cross-validation on training data since EVALITA does not provide a validation set. In particular, we perform optimization in order to choose the best optimization algorithm evaluating among Adadelta, Adagrad, Adam and SGD. Regarding SGD, we test several values of the initial learning

rate in the set {0.01, 0.0125, 0.15} and values for the decay rate in the set {0.01, 0.05, 0.1}. Moreover, we optimize the number of epochs (we set the maximum number of epochs to 60).

Results of the optimization procedure are reported in Table 2. Results about SGD parameters are removed since they give rise to lower performance.

Table 2
Results of the optimization

Task	Opt.Alg.	Epochs
PoS-tagging	Adadelta	60
NER	Adadelta	57
Supersense	Adagrad	60
Multi-task	Adam	60

3.1 PoS-tagging of Tweets

The goal of the task is to perform PoS-tagging of tweets. The task is more challenging with respect to the canonical PoS-tagging task due to the short and noisy nature of tweets. For the evaluation we adopt the dataset used during the EVALITA 2016 PoSTWITA task (Bosco et al. 2016) in order to compare our system with the other EVALITA participants. The dataset contains 6,438 tweets (114,967 tokens) for training and 300 tweets (4,759 tokens) for testing. A training sample is reported in Figure 4, where each token is annotated with its PoS-tag. The metric used for the evaluation is the classical tagging accuracy defined as the number of correct PoS-tag assignments divided by the total number of tokens in the test set. Participants can predict only one tag for each token.

```
Governo  Monti   :      decreto  in    cdm    per  approvazione   !    http://t.co/Z76KLLGP
PROPN    PROPN  PUNCT  NOUN     ADP   PROPN  ADP  NOUN           PUNCT  URL
```

Figure 4
A PoS-tagging training sample.

All the top-performing PoSTWITA systems are based on Deep Neural Networks and, in particular, on LSTM. Moreover, most systems use word or character embeddings as inputs for their systems. This makes other systems more similar to the one proposed in this paper.

Results of the evaluation are reported in Table 3. Our approach (*DL-ita*) is able to provide results in line with the first three PoSTWITA participants. In (Basile, Semeraro, and Cassotti 2017), we report an accuracy of .9334 using the same system, but running 100 epochs. In this work, we set the maximum number of epochs to 60 during the optimization step in order to reduce the computation time. Nevertheless, results prove the effectiveness of the proposed architecture without exploiting task/language specific resources. The only used resource is a corpus of 70M tweets randomly extracted from Twita, a collection of about 800M tweets, for building the word embeddings.

Table 3
Results for the PoSTWITA task.

System	Accuracy
DL-ita	.9265
ILC-CNR	.9319
UniDuisburg	.9286
UniBologna UnOFF	.9279

It is important to underline that the best system (*ILC-CNR*) (Cimino and Dell'Orletta 2016) in PoSTWITA uses a bi-LSTM and an RNN by exploiting both word and character embeddings, moreover it uses further features based on morpho-syntactic categories and spell checker. The second best system (*UniDuisburg*) (Horsmann and Zesch 2016) in PoSTWITA exploits a CRF classifier using several features without a DL architecture, while the system *UniBologna UnOFF* (Tamburini 2016) uses a bi-LSTM with a CRF layer by exploiting word embeddings and additional morphological features.

3.2 NER Task

Three tasks about named entities have been organized during the EVALITA evaluation campaigns, respectively in 2007 (Speranza 2007), 2009 (Speranza 2009), and 2011 (Lenzi, Speranza, and Sprugnoli 2013). In this paper we take into account the 2009 edition since the I-CAB dataset [9] used in the evaluation is the same adopted in 2009. In 2007 a different version of I-CAB was used, while in 2011 the task was focused on data transcribed by an ASR system. The I-CAB dataset consists of a set of news manually annotated with four kinds of entities: GPE (geo-political), LOC (location), ORG (organization) and PER (person). The dataset contains 525 news for training and 180 for testing for a total number of 11,410 annotated entities for training and 4,966 ones for testing. The dataset is provided in the IOB2 format: the tag B (for "begin") denotes the first token of a Named Entity, I (for "inside") is used for all other tokens in a Named Entity, and O (for "outside") is used for all other words. The Entity type tags are: PER (for Person), ORG (for Organization), GPE (for GeoPolitical Entity), or LOC (for Location). A training sample is reported in Figure 5.

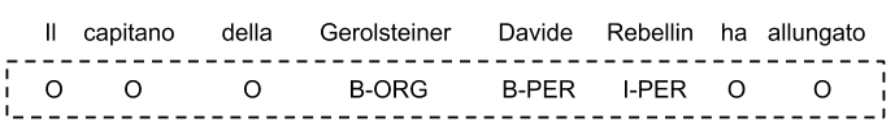


Figure 5
A NER training sample.

9 http://ontotext.fbk.eu/icab.html

We build word embeddings by exploiting the Italian version of Wikipedia. Word2vec (Mikolov et al. 2013) is used for creating embeddings with a dimension of 300; we remove all words that have less than 40 occurrences in Wikipedia. For the other parameters, we adopt the standard values provided by word2vec.

Table 4
Results for the Italian NER task compared with other EVALITA 2009 participants.

System	ALL			GPE	LOC	ORG	PER
	P	R	F1	F1	F1	F1	F1
DL-ita	.8236	.8197	.8217	.8579	.5905	.6673	.9203
FBK_ZanoliPianta	.8407	.8002	.8200	.8513	.5124	.7056	.8831
UniGen_Gesmundo_r2	.8606	.7733	.8146	.8336	.5081	.7108	.8741
UniTN-FBK-RGB_r2	.8320	.7908	.8109	.8525	.5224	.6961	.8689

Results of the evaluation are reported in Table 4, where our system (*DL-ita*) is compared with respect to the other EVALITA 2009 participants. The system outperforms the first three EVALITA participants thanks to the best performance in recall. All the first three participants adopt classical classification methods: the first system (Zanoli, Pianta, and Giuliano 2009) combines two classifiers (HMM and CRF), the second participant (Gesmundo 2009) uses a Perceptron algorithm, while the third (Mehdad, Scurtu, and Stepanov 2009) adopts Support Vector Machine and feature selection. We can conclude that the DL architecture is more effective in model generalization and in tackling the data sparsity problem. This behavior is supported by the good performance in recognizing LOC entities. In fact, the LOC class represents about the 3% of annotated entities in both training and test.

Other two systems (Nguyen and Moschitti 2012; Bonadiman, Severyn, and Moschitti 2015) able to overcome the EVALITA 2009 participants have been proposed in the literature. The former (Nguyen and Moschitti 2012) achieves the 84.33% of F1 by using re-ranking techniques and the combination of two state of the art NER learning algorithms: conditional random fields and support vector machines. The latter (Bonadiman, Severyn, and Moschitti 2015) exploits a Deep Neural Network with a log-likelihood cost function and a recurrent feedback mechanism to ensure the dependencies between the output tags. This system is able to achieve the 82.81% of F1, a performance comparable with our DL architecture.

3.3 Super Sense Tagging

The Super-Sense Tagging (SST) task (Dei Rossi, Di Pietro, and Simi 2011) consists in annotating each significant entity in a text, like nouns, verbs, adjectives and adverbs, within a general semantic taxonomy defined by the WordNet lexicographer classes (called super-senses, for a total of 45 senses). SST can be considered as a task half-way between NER and Word Sense Disambiguation (WSD). It is an extension of NER since it uses a larger set of semantic categories, and it is an easier and more practical task with respect to WSD. The dataset has been tagged using the IOB2 format as for the NER task and contains about 276,000 tokens for training and about 50,000 for testing. A training sample is reported in Figure 6, where each token occurring in WordNet is annotated with its super sense. The metric adopted for the evaluation is the F1. Results of the evaluation are reported in Table 5.

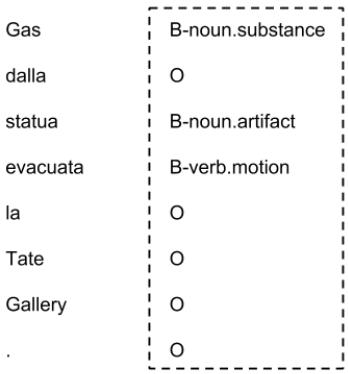


Figure 6
A SST training sample.

As word embeddings, we use the same ones adopted for the NER task and built upon Wikipedia with lowercase. Moreover, we exploit PoS-tags as additional features.

Table 5
Results for the Super-Sense Tagging task.

System	F1
DL-ita	.7864
UNIBA-SVMcat	.7866
UNIPI-run3	.7827

Our system (*DL-ita*) is very close to the best system in EVALITA 2011 SST task *UNIBA-SVMcat*. This system combines lexical and distributional features through an SVM classifier, in particular it exploits specific features such us: lemma, contextual PoS-tags, the super-sense assigned to the most frequent sense of the word and information about the grammatical conjugation of verbs. We plan to introduce this kind of features into the DL system in order to understand if this difference in performance still emerges. The second system (*UNIPI-run3*) (Attardi et al. 2011) exploits lexical features and a Maximum Entropy classifier.

3.4 Multi-task Evaluation

In this evaluation, we exploit shared data coming from EVALITA 2016. In particular, we use PoS-tagging and sentiment analysis data. We choose these two tasks because they share the largest training set. We plan to investigate more annotation layers when the number of shared examples in training/testing set will increase. PoS-tagging data have been previously described in Section 3.1, while sentiment data are taken from SENTIPOLC (Barbieri et al. 2016). SENTIPOC (*SENTIment POLarity Classification*) is a sentiment analysis task where systems are required to automatically annotate tweets with a tuple of boolean values indicating the message's subjectivity, its polarity (positive or negative), and whether it is ironic or not. For example, the following tweet "*Dopo due*

ore che stavo studiando italiano, mi sono accorta che avevo preso il libro sbagliato. #benecosi"[10] is annotated with the subjectivity, positive and irony tags.

The SENTIPOLC training set consists of 7,410 tweets (6,412 are shared with the PoSTWITA task), while the test set contains 2,000 tweets (300 are shared with PoSTWITA). In conclusion, we are able to train the multi-task architecture using 6,412 tweets, while the accuracy of PoS-tag is evaluated on 300 tweets and the performance on SENTIPOLC is computed on 2,000 tweets.

Table 6
Results for the PoSTWITA task using the multi-task architecture.

System	Accuracy
DL-ita-MT	.9246
ILC-CNR	.9319
UniDuisburg	.9286
UniBologna UnOFF	.9279
DL-ita	.9265

Table 6 reports the accuracy on PoS-Twita. The multi-task architecture (DL-ita-ML) obtains results similar to the single-task learning architecture (*DL-ita*). Results show that PoS-tag is not able to exploit information about polarity and irony for improving performance.

Table 7
Results for the SENTIPOLC task

System	Subj.	Positive	Negative	Total	Irony
DL-ita-ML	.7176	.6361	.6521	.6441	.5120
DL-ita	.7282	.6391	.6802	.6602	.4970
Unitor.1.u	**.7444**	.6354	.6885	.6620	.4728
Unitor.2.u	.7351	.6312	.6838	.6575	.4810
samskara.1.c	.7184	.5198	.6168	.5683	-
UniPI.2.c	.6937	.6850	.6426	**.6638**	-
tweet2check16.c	.6236	.6153	.5878	.6016	**.5412**
CoMoDI.c -	-	-	-	-	.5251
tweet2check14.c	.5843	.5660	.6034	.5847	.5162

Regarding the SENTIPOLC task, results in Table 7 show that the multi-task architecture is able to improve its performance in the irony task. However, performance decreases in the polarity task. In conclusion, information about the PoS-tag is useful for irony, but not in the subjective and polarity tasks. It is not easy to interpret the DL architecture and this is made even more difficult by the multi-task learning. For example, the following tweet was correctly classified by the DL-ita-ML and incorrectly

10 In English: "After two hours I was studying Italian, I realized that I had taken the wrong book. #welldone"

classified by the DL-ita: *Io mi lamento della gente che scrive ancora "freddy mercury" ma anche quella che scrive "jhonny cash" non scherza*[11].

Moreover, Table 7 reports results for the systems at the intersection between the first three systems of each SENTIPOLC subtask. Our system (DL-ita-ML) is able to achieve good results in each subtask and ranks 4 out of 18 in the subjective task, 6 out of 25 in the polarity task and 5 out of 12 in the irony task.

The best system in the subjective task, Unitor1.u (Castellucci, Croce, and Basili 2016), reports also good performance in the polarity task but poor performance in the irony task. In particular, Unitor1.u implements a workflow of several Convolutional Neural Networks classifiers in which sentiment specific information is injected using Polarity Lexicons (Basili, Croce, and Castellucci 2017) automatically acquired through the analysis of unlabeled collection of tweets. Conversely, our system does not exploit any additional resources.

The best system in the polarity task, UniPI.2 (Attardi et al. 2016), adopts Convolutional Neural Networks as *Unitor1.u* and exploits both word embeddings and Sentiment Specific word embeddings. This system ranks eighth in the subjective task and does not participate in the irony one.

Finally, the best system in the irony task, tweet2check16.c (Di Rosa and Durante 2016), is an industrial system which combines many different classifiers, each of which is built by using different machine learning algorithms and implementing different features. This system is able to achieve moderate performance in the polarity task, while poor performance are reported in the subjective task.

The CoMoDI.c (Frenda 2016) is a rule based system specifically developed for irony detection and it is able to achieve good performance[12] in the irony task. The samskara system (Russo and Monachini 2016) is the third in the final rank of the subjective task. This system uses a Naive Bayes classifier trained on a set of structural features specifically designed for the Twitter domain. However, this system achieves the worst performance in the final rank of the polarity task and it does not participate in the irony task.

In conclusion, the multi-task architecture obtains good performance in all SENTIPOLC subtasks. However, it does not achieve the best performance in any specific task. Moreover, we observe that the information about PoS-tags is useful in the irony subtask, while the use of polarity information in the PoS-tag results in a slight decrease in accuracy.

4. Conclusions and Future Work

We propose an evaluation of a state of the art DL architecture in the context of the Italian language. In particular, we consider three sequence labeling tasks: PoS-tagging of tweets, Named Entity Recognition and Super-Sense Tagging. We also propose a multi-task learning architecture involving PoS-tagging and sentiment analysis. All tasks exploit data coming from EVALITA, a standard benchmark for the evaluation of Italian NLP systems.

Our system is able to achieve good performance in all the tasks without using handcrafted features. Analyzing the results, we observe that our system is able to achieve

11 In English: *I complain about the people who still write "freddy mercury" but also the one who writes "jhonny cash" is not joking.*

12 second in the final rank

state of the art performance for the Italian language in all the sequence labeling tasks. This proves the effectiveness of the DL architecture in a different language - in this case Italian - without using language specific features

Regarding the multi-task learning, our architecture is able to achieve good performance in each subtask (subjectivity, polarity and irony) using the same architecture. In addition, the multi-task learning results show that the irony task benefits from the information provided by PoS-tags. In this work, we are able to investigate only PoS-tagging and sentiment analysis because they share the largest training set in EVALITA. We plan to investigate more annotation layers when the number of shared examples in training/testing set will increase.

As future work, we plan to investigate further multi-task learning architectures exploiting different strategies, such as the one proposed in (Hashimoto et al. 2016), where higher layers include short-cut connections to lower-level task predictions to reflect linguistic hierarchies.

Acknowledgments

This work is partially supported by the project "Multilingual Entity Liking" funded by the Apulia Region under the program FutureInResearch.

References

Attardi, Giuseppe, Luca Baronti, Stefano Dei Rossi, and Maria Simi. 2011. SuperSense Tagging with a Maximum Entropy Classifier and Dynamic Programming. In *Working Notes of EVALITA 2011*.

Attardi, Giuseppe, Daniele Sartiano, Chiara Alzetta, and Federica Semplici. 2016. Convolutional Neural Networks for Sentiment Analysis on Italian Tweets. In Pierpaolo Basile, Anna Corazza, Franco Cutugno, Simonetta Montemagni, Malvina Nissim, Viviana Patti, Giovanni Semeraro, and Rachele Sprugnoli, editors, *Proceedings of Third Italian Conference on Computational Linguistics (CLiC-it 2016) & Fifth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2016)*, Naples, Italy. aAccademia University Press.

Barbieri, Francesco, Valerio Basile, Danilo Croce, Malvina Nissim, Nicole Novielli, and Viviana Patti. 2016. Overview of the EVALITA 2016 SENTiment POLarity Classification Task. In Pierpaolo Basile, Anna Corazza, Franco Cutugno, Simonetta Montemagni, Malvina Nissim, Viviana Patti, Giovanni Semeraro, and Rachele Sprugnoli, editors, *Proceedings of Third Italian Conference on Computational Linguistics (CLiC-it 2016) & Fifth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2016)*, Naples, Italy. aAccademia University Press.

Basile, Pierpaolo, Franco Cutugno, Malvina Nissim, Viviana Patti, and Rachele Sprugnoli. 2016. EVALITA 2016: Overview of the 5th evaluation campaign of natural language processing and speech tools for Italian. In Pierpaolo Basile, Anna Corazza, Franco Cutugno, Simonetta Montemagni, Malvina Nissim, Viviana Patti, Giovanni Semeraro, and Rachele Sprugnoli, editors, *Proceedings of Third Italian Conference on Computational Linguistics (CLiC-it 2016) & Fifth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2016)*, Naples, Italy. aAccademia University Press.

Basile, Pierpaolo, Giovanni Semeraro, and Pierluigi Cassotti. 2017. Bi-directional LSTM-CNNs-CRF for Italian Sequence Labeling. In Roberto Basili, Malvina Nissim, and Giorgio Satta Satta, editors, *Proceedings of the Fourth Italian Conference on Computational Linguistics (CLiC-it 2017)*. Associazione Italiana di Linguistica Computazionale (AILC).

Basili, Roberto, Danilo Croce, and Giuseppe Castellucci. 2017. Dynamic polarity lexicon acquisition for advanced social media analytics. *International Journal of Engineering Business Management*, 9:1–18.

Bonadiman, Daniele, Aliaksei Severyn, and Alessandro Moschitti. 2015. Deep neural networks for named entity recognition in italian. In *CLiC-it 2015 Proceedings of the second Italian Conference on Computational Linguistics*, page 51.

Bosco, Cristina, Fabio Tamburini, Andrea Bolioli, and Alessandro Mazzei. 2016. Overview of the EVALITA 2016 Part of speech on twitter for Italian task. In Pierpaolo Basile, Anna Corazza, Franco Cutugno, Simonetta Montemagni, Malvina Nissim, Viviana Patti, Giovanni Semeraro, and Rachele Sprugnoli, editors, *Proceedings of Third Italian Conference on Computational Linguistics (CLiC-it 2016) & Fifth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2016)*, Naples, Italy. aAccademia University Press.

Castellucci, Giuseppe, Danilo Croce, and Roberto Basili. 2016. Context-aware Convolutional Neural Networks for Twitter Sentiment Analysis in Italian. In Pierpaolo Basile, Anna Corazza, Franco Cutugno, Simonetta Montemagni, Malvina Nissim, Viviana Patti, Giovanni Semeraro, and Rachele Sprugnoli, editors, *Proceedings of Third Italian Conference on Computational Linguistics (CLiC-it 2016) & Fifth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2016)*, Naples, Italy. aAccademia University Press.

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

Cimino, Andrea and Felice Dell'Orletta. 2016. Building the state-of-the-art in POS tagging of Italian Tweets. In Pierpaolo Basile, Anna Corazza, Franco Cutugno, Simonetta Montemagni, Malvina Nissim, Viviana Patti, Giovanni Semeraro, and Rachele Sprugnoli, editors, *Proceedings of Third Italian Conference on Computational Linguistics (CLiC-it 2016) & Fifth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2016)*, Naples, Italy. aAccademia University Press.

Dei Rossi, Stefano, Giulia Di Pietro, and Maria Simi. 2011. EVALITA 2011: Description and Results of the SuperSense Tagging Task. In *Working Notes of EVALITA 2011*.

Di Rosa, Emanuele and Alberto Durante. 2016. Tweet2Check evaluation at Evalita Sentipolc 2016. In Pierpaolo Basile, Anna Corazza, Franco Cutugno, Simonetta Montemagni, Malvina Nissim, Viviana Patti, Giovanni Semeraro, and Rachele Sprugnoli, editors, *Proceedings of Third Italian Conference on Computational Linguistics (CLiC-it 2016) & Fifth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2016)*, Naples, Italy. aAccademia University Press.

Frenda, Simona. 2016. Computational rule-based model for Irony Detection in Italian Tweets. In Pierpaolo Basile, Anna Corazza, Franco Cutugno, Simonetta Montemagni, Malvina Nissim, Viviana Patti, Giovanni Semeraro, and Rachele Sprugnoli, editors, *Proceedings of Third Italian Conference on Computational Linguistics (CLiC-it 2016) & Fifth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2016)*, Naples, Italy. aAccademia University Press.

Gesmundo, Andrea. 2009. Bidirectional sequence classification for named entities recognition. In *Poster and Workshop Proceedings of the 11th Conference of the Italian Association for Artificial Intelligence*.

Hashimoto, Kazuma, Caiming Xiong, Yoshimasa Tsuruoka, and Richard Socher. 2016. A joint many-task model: Growing a neural network for multiple nlp tasks. *arXiv preprint arXiv:1611.01587*.

Horsmann, Tobias and Torsten Zesch. 2016. Building a social media adapted PoS tagger using flexTag - A case study on Italian tweets. In Pierpaolo Basile, Anna Corazza, Franco Cutugno, Simonetta Montemagni, Malvina Nissim, Viviana Patti, Giovanni Semeraro, and Rachele Sprugnoli, editors, *Proceedings of Third Italian Conference on Computational Linguistics (CLiC-it 2016) & Fifth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2016)*, Naples, Italy. aAccademia University Press.

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

Lenzi, Valentina Bartalesi, Manuela Speranza, and Rachele Sprugnoli. 2013. Named entity recognition on transcribed broadcast news at EVALITA 2011. In *Revised Papers from EVALITA 2011: International Workshop on the Evaluation of Natural Language and Speech Tools for Italian*, volume 7689, pages 86–97. Springer.

Luo, Gang, Xiaojiang Huang, Chin-Yew Lin, and Zaiqing Nie. 2015. Joint named entity recognition and disambiguation. In *Conference Proceedings - EMNLP 2015: Conference on Empirical Methods in Natural Language Processing*, pages 879–888.

Ma, Xuezhe and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. *arXiv preprint arXiv:1603.01354*.

Magnini, Bernardo and Amedeo Cappelli. 2009. Introduction to Evalita 2009. In *Poster and Workshop Proceedings of the 11th Conference of the Italian Association for Artificial Intelligence*.

Mehdad, Yashar, Vitalie Scurtu, and Evgeny Stepanov. 2009. Italian named entity recognizer participation in NER task @ Evalita 09. In *Poster and Workshop Proceedings of the 11th Conference of the Italian Association for Artificial Intelligence*.

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Nguyen, Truc-Vien T. and Alessandro Moschitti. 2012. Structural reranking models for named entity recognition. *Intelligenza Artificiale*, 6(2):177–190.

Passos, Alexandre, Vineet Kumar, and Andrew McCallum. 2014. Lexicon infused phrase embeddings for named entity resolution. *arXiv preprint arXiv:1404.5367*.

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

Ruder, Sebastian. 2017. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*.

Russo, Irene and Monica Monachini. 2016. Samskara Minimal structural features for detecting subjectivity and polarity in Italian tweets. In Pierpaolo Basile, Anna Corazza, Franco Cutugno, Simonetta Montemagni, Malvina Nissim, Viviana Patti, Giovanni Semeraro, and Rachele Sprugnoli, editors, *Proceedings of Third Italian Conference on Computational Linguistics (CLiC-it 2016) & Fifth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2016)*, Naples, Italy. aAccademia University Press.

Speranza, Manuela. 2007. Evalita 2007: the named entity recognition task. In *Proceedings of the Workshop Evalita 2007*.

Speranza, Manuela. 2009. The named entity recognition task at Evalita 2009. In *Poster and Workshop Proceedings of the 11th Conference of the Italian Association for Artificial Intelligence*.

Srivastava, Nitish, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.

Tamburini, F. 2016. A BiLSTM-CRF PoS-tagger for Italian tweets using morphological information. In Pierpaolo Basile, Anna Corazza, Franco Cutugno, Simonetta Montemagni, Malvina Nissim, Viviana Patti, Giovanni Semeraro, and Rachele Sprugnoli, editors, *Proceedings of Third Italian Conference on Computational Linguistics (CLiC-it 2016) & Fifth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2016)*, Naples, Italy. aAccademia University Press.

Zanoli, Roberto, Emanuele Pianta, and Claudio Giuliano. 2009. Named entity recognition through redundancy driven classifiers. In *Poster and Workshop Proceedings of the 11th Conference of the Italian Association for Artificial Intelligence*.

Zhang, Yu and Qiang Yang. 2017. A survey on multi-task learning. *arXiv preprint arXiv:1707.08114*.

Multitask Learning with Deep Neural Networks for Community Question Answering

Daniele Bonadiman*
Università di Trento

Antonio Uva**
Università di Trento

Alessandro Moschitti†
Amazon

In this paper, we developed a deep neural network (DNN) that learns to solve simultaneously the three tasks of the cQA challenge proposed by the SemEval-2016 Task 3, i.e., question-comment similarity, question-question similarity and new question-comment similarity. The latter is the main task, which can exploit the previous two for achieving better results. Our DNN is trained jointly on all the three cQA tasks and learns to encode questions and comments into a single vector representation shared across the multiple tasks. The results on the official challenge test set show that our approach produces higher accuracy and faster convergence rates than the individual neural networks. Additionally, our method, which does not use any manual feature engineering, approaches the state of the art established with methods that make heavy use of it.

1. Introduction

Community Question Answering (cQA) websites enable users to freely ask questions in web forums and expect some good answers in the form of comments from the other users. Given the large number of question/answer pairs available on cQA sites, researchers started to investigate the possibility to exploit user-generated content for training automatic QA systems. Unfortunately, the text involved in the cQA scenario is rather noisy, therefore, providing models that outperform the simple bag-of-words representation can result rather difficult. The challenge, SemEval-2016 Task 3 "Community Question Answering", has been designed to study the above problems: the participants were supposed to build a fully automatic system for cQA. In particular, given a fresh user question, q_{new}, and a set of forum questions, Q, answered by a comment set, C, the main task consists of determining whether a comment $c \in C$ is a pertinent answer of q_{new} or not. This task can be divided into three sub-tasks:

(A) predict if a comment produced in response to a question contains a valid answer;

* Dept. of Information Engineering and Computer Science (DISI) - Via Sommarive, 9, 38123 Povo, Trento, Italy. E-mail: d.bonadiman@unitn.it
** Dept. of Information Engineering and Computer Science (DISI) - Via Sommarive, 9, 38123 Povo, Trento, Italy. E-mail: antonio.uva@unitn.it
† Manhattan Beach, CA, USA, 90266. E-mail: amosch@amazon.com. This work was carried out when the author was at the University of Trento.

(B) re-rank a set of questions according to their relevancy with respect to the original question; and

(C) predict if a comment produced in response to a previous question posed on the cQA forum represents a valid answer to a fresh question.

Traditionally, these tasks have been tackled by designing systems/classifiers that target each task separately. Each classifier accepts a vector encoding a text pair (e.g., a question/question or a question/answer pair) in input by using many complex lexical syntactic or semantic features and, then, computing similarity between these representations. However, this approach suffers from the drawbacks of requiring a "customized" set of features for each task being solved.

Recent work on deep neural networks (DNNs) for Multitask Learning (MTL) (Collobert and Weston 2008; Liu et al. 2015) showed that is possible to *jointly train* a general system that solves different tasks simultaneously. Inspired by the success of MTL, in this paper, we propose a DNN model that leverages the data from the three cQA tasks of SemEval. Indeed, as the three tasks are highly related, we claim that cQA can benefit from this approach. We show that, despite the fact that does not require any feature engineering, our DNN can approach the performance of the best systems, which use heavy feature engineering. Additionally, we are going to make the corpora for studying MTL on this interesting challenge available to the research community.

2. cQA Tasks at SemEval

The research problem issued by SemEval-2016 Task 3 is exemplified by Fig. 2: given a new question q_{new}, Task C is about directly retrieving a relevant comment from the entire community. This can also be achieved by solving Task B, which finds a similar question, q_{rel}, and then executing Task A, which selects good comments, c_{rel}, for q_{rel}. It should be noted that Task A classifies comments, specifically written by the users for q_{rel}, whereas Task C classifies comments written by the users for other, sometimes, similar questions. This means, it needs to filter out comments that can be partially related to q_{new} (because they correctly answer the related question, q_{rel}) but still not correctly answering q_{new}. Clearly, Task C classifier needs to tackle a much more semantically challenging task. Thus, tasks A and C are semantically and computationally rather different and together with Task B: they constitute an interesting MTL problem since differences and correlations are played at a very high semantic level.

2.1 Task A: Question-Comment Similarity

Given a question, q_{rel}, and its first 10 comments, c_{rel}, in the question threads, rerank the comments according to their relevance to q_{rel}. Relevancy is defined according to three classes: (*i*) `good`: the comment is definitively relevant; (*ii*) `potentially useful`: the comment is not good, but it still contains related information worth checking; and (*iii*) `bad`: the comment is irrelevant (e.g., it is part of a dialogue or unrelated to the topic). For evaluation purposes, both `potentially useful` and `bad` comments were considered as `bad`.

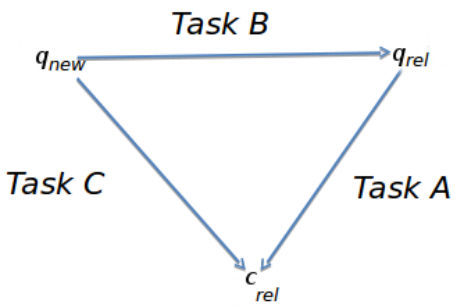


Figure 1
The 3 tasks of cQA at SemEval: the arrows show the relations between the original and the related questions and the related comments.

2.2 Task B: Question-Question Similarity

Given a new question, q_{new}, and its first 10 related questions (retrieved by a search engine), q_{rel}, rerank them according to their similarity with respect to q_{new}. Relevancy is expressed by three classes: (*i*) `perfect match`: the new and forum questions request roughly the same information, (*ii*) `relevant`: the new and forum questions ask for similar information, or (*iii*) `irrelevant`: the new and forum questions are completely unrelated. For evaluation purposes, both `perfect match` and `relevant` forum questions are considered as `relevant`.

2.3 Task C: New Question-Comment Similarity

Given a new question, q_{new}, and its first 10 related questions (retrieved by a search engine), q_{rel}, each associated with its first 10 comments, c_{rel}, appearing in its thread, rerank the 100 comments (10 questions $\times$ 10 comments) according to their relevance with respect to q_{new}. Relevancy is defined similarly to task A.

2.4 Dataset

The data for the above-mentioned tasks is distributed in three datasets: train, dev and test sets. The distribution of questions and comments in each dataset varies across the different tasks: Task A contains 6,938 related questions and 40,288 comments. Each comment in the dataset was annotated with a label indicating its relevancy with respect to the related question. Task B contains 317 original questions. For each original question, 10 related questions were retrieved, summing to 3,169 related questions. Also in this case, the related questions were annotated with a relevancy label, which tells if they are relevant with respect to the user original question. Task C contains 317 original questions, together with 3,169 related questions (same as in Task B) and 31,690 comments. Each comment was labelled with its relevancy with respect to the original question.

3. A General Deep Architecture for cQA

All the previous tasks are about reranking questions or comments with respect to an original question. In the following, we describe a general architecture for solving them.

3.1 Deep Architecture for relational learning from pairs of text

A traditional approach to cQA is to learn a different classifiers for solving each of these three tasks, independently. For example, first a classifier can be trained to rerank a set of related questions retrieved by a search engine, using their similarity with respect to the user question (Task B). Then, another classifier can be trained to rerank the list of comments appearing in the threads of similar questions (Task A). Each of these classifiers uses a different set of task-dependent features. In this work, we use a neural network architecture for classifying text pairs. The network is fed using the different pairs, (q_{rel}, c_{rel}), (q_{new}, q_{rel}) and (q_{new}, c_{rel}), to learn the tasks A, B and C, respectively, and produces a similarity score that can be used to rerank questions or comments.

It is composed of two main components: (i) two sentence encoders that map input sentences i into fixed size vectors $x_{s_i} \in \mathbb{R}^m$, and (ii) a feed forward neural network that computes the similarity between these two sentence vectors.

The sentence encoders are composed of (i) a sentence matrix $\mathbf{s}_i \in \mathbb{R}^{d \times |\mathbf{i}|}$, where d is the size of the word embeddings, obtained by concatenating the vectors of the corresponding words in the input sentence $\mathbf{w}_j \in \mathbf{s}_i$, and (ii) a sentence model $f : \mathbb{R}^{d \times |\mathbf{i}|} \rightarrow \mathbb{R}^m$, which maps the sentence matrix to a fixed size sentence embedding $x_{s_i} \in \mathbb{R}^m$.

The choice of the sentence model plays a crucial role as the resulting intermediate representation of the input sentences affects the successive steps of computing their similarity. Previous work in this direction uses different types of sentence models such as LSTM, distributional sentence model (average of word vectors), and convolutional sentence model. In particular, the latter is composed of a sequence of convolution and pooling feature maps, which have achieved the state of the art in various NLP tasks (Kalchbrenner, Grefenstette, and Blunsom 2014; Kim 2014).

In this paper, we used a CNN sentence model generated by a convolutional operation followed by a k-max pooling layer with $k = 1$, since it provides comparable performance to the LSTM on the task of new question-comment similarity, as shown in Table 2. The sentence encoder, $x_{s_i} = f(s_i)$, outputs a fixed-size embedding of the input sentence s_i. The sentence vectors, x_{s_i}, are concatenated together and given in input to a Multi-Layer Perceptron, which is constituted by a non-linear hidden layer and an sigmoid output layer.

3.2 Injecting Relational Information

All the tasks we consider require to model relations between words present in the two pieces of text. For this purpose, we encode the relation in forms of discrete features, as described in (Collobert et al. 2011), i.e., using an additional embedding layer. They augmented the word embedding with the corresponding feature embedding. Thus, given a word, w_j, its final word embedding is defined as $\mathbf{w}_j \in \mathbb{R}^d$, where $d = d_w + d_{feat}$, where d_w is the size of the word embedding and d_{feat} is the size of the feature embedding.

We use a discrete feature, represented with an embedding of 5 dimensions, to encode matches between two words in the two input pieces of text. In particular, we associate each word w in the input sentences with a *word overlap* index $o \in \{0, 1\}$, where $o = 1$ means that w is shared by both Q and C (or by the two questions for task

B), i.e., overlaps, $o = 0$ otherwise. It should be noted that the embeddings described here cannot be considered as task specific features, manually handcrafted. They are part of the network, serve the purpose of injecting relational information between the representations of the two input texts and can be generally applied to different domains, data and tasks.

3.3 Adding the rank features

The SemEval problems concern reranking text initially ranked by Google and made available to the participants for tasks B and C. Considering that the Google rank is computed using powerful algorithms and a lot of resources, it is essential to encode it in our networks. There are several methods to achieve this. After some experiments, we opted for discretizing the rank values in 5 different bins of different sizes, i.e. $[1-2], [2-5], [5-10], [10-25], [25-\infty]$. The rank feature is added to the joint layer, where the output of the sentence model is concatenated, using a table lookup operation. It should be noted that for each task, we use a different relation feature (overlap embeddings) between each pair of text.

4. MTL for cQA

MTL aims at learning several related tasks at the same time to improve some (or possibly all) tasks using joint information (Caruana 1997). MTL is particularly well suited for modeling Task C as it is a composition of tasks A and B, thus, it can benefit from having both questions q_{new} and q_{rel} as input to better model the interaction between the new question and the comment. More precisely, it can use the triplet $\langle q_{new}, q_{rel}, c_{rel} \rangle$ in the learning process, where the interaction between the triplet members is exploited during the joint training of the three models of the tasks, A, B and C. In fact, an improvement on question-comment similarity or on question-question similarity can lead to an improvement in the task of new question-comment similarity (Task C).

Additionally, each thread in the SemEval dataset is annotated with the labels for all the three tasks and therefore it is possible to apply joint learning directly.

4.1 Joint Learning Architecture

Our Joint learning architecture is depicted in Figure 2, it is a direct extension of the architecture proposed for Task C (Section 3.3). It takes the three sentences as input, i.e, a new question, q_{new}, the related question, q_{rel}, and its comment, c_{rel}, and produces three fixed size representations, $x_{q_{new}}$, $x_{q_{rel}}$ and $x_{c_{rel}}$, respectively.

These three representations are then concatenated ($h_j = [x_{q_{new}}, x_{q_{rel}}, x_{r_{rel}}]$) and fed to a hidden layer to create a shared representation of the input for the three tasks, $h_s = Wh_j$.

The output of this layer, h_s is then fed to three independent Multilayer Perceptrons (MLP) that produce the scores for the three tasks. To directly apply MTL, we use the binary cross-entropy instead of the max margin loss as our objective function. The main motivation is that such function is computed based on pairs of positive-negative examples that cannot be created with multiple labels. At training time, for each example, the loss is calculated on the three outputs of the network. The final loss is then the sum of the individual losses for the three tasks.

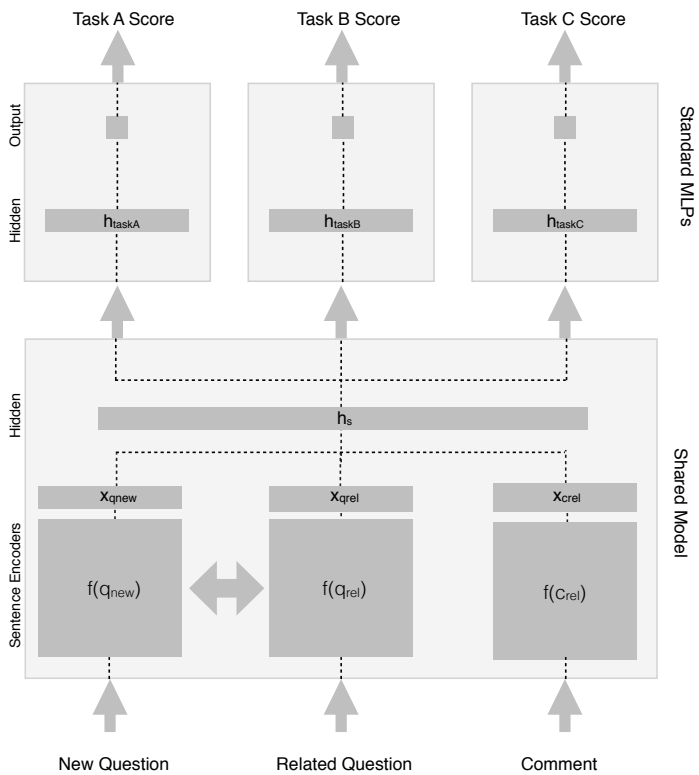


Figure 2
Our MTL architecture, where the three sentences are at the bottom. They pass through the sentence encoders. The output is concatenated and fed to a hidden layer whose output is passed to three independent multi-layer perceptrons, which produce the scores for the individual tasks. The double arrow, $\leftrightarrow$, indicates a shared sentence model between q_{new} and q_{rel}.

4.2 Shared Sentence Models

The SemEval dataset contains ten times less new questions than related questions by construction. However, all questions, q_{new} included, are supposed to be of the same nature. Thus we can certainly use a shared text model for modeling better representations for both new and related questions. Formally, let $x_d = f(d, \theta)$ be a sentence model for document d with parameters θ, i.e., the embedding weights and the convolutional filters. In our original formulation, each sentence model uses a different set of parameters $\theta_{q_{new}}$, $\theta_{q_{rel}}$ and $\theta_{c_{rel}}$. However, for the question representation, we also used the same set of parameters θ_q. Such shared sentence model is illustrated by a double arrow in in Figure 2.

5. Experiments

5.1 Setup

We encode input sentences with fixed-sized vectors using a convolutional operation of size 5 and a k-max pooling operation with $k = 1$, i.e., similarly to (Severyn and Moschitti

Table 1
Percentage of positive examples in the training datasets for each task.

	Task A	Task B	Task C
Train	37.51%	39.41%	9.9%
Train + ED	37.47%	64.38%	21.25%

2015, 2016). We use two non-linear hidden layers (with hyperbolic tangent activation, Tanh), whose size is equal to the size of the previous layer, i.e., the join layer. We include information such as word overlaps and rank position as embedding with an additional lookup table with vectors of size $d_{feat} = 5$.

Pre-processing: both questions and comments are tokenized and lowercased (to reduce the dimensionality of the dictionary and therefore of the embedding matrix). Moreover, question subject and body are concatenated to create a unique question. For computational reasons, we opted to limit the size of the input text at 100 words: we did not observe any degradation in performance.

Word Embeddings: for all the proposed models, we pre-initialize the word embedding matrices with standard skipgram embedding of dimensionality 50 trained on the English Wikipedia dump using word2vec toolkit (Mikolov et al. 2013).

Training: The network is trained using SGD with shuffled mini-batches using the rmsprop update rule (Tieleman and Hinton 2012). The model learns until the validation loss stops improving, with patience $p = 10$, i.e., the number of epochs to wait before early stopping, if no progress on the validation set is obtained. In fact, early stopping (Prechelt 1998) allows us to avoid overfitting and improving the generalization capabilities of the network. For the MTL architecture, we employed two different stopping criteria. The first is to stop training when the global validation loss does not improve anymore (the sum of the individual losses of the three tasks). The second, instead, saves three different models and evaluates them when the individual losses stop improving. Since the three tasks converge at different epochs, the first method may lead to sub-optimal results for the individual tasks, but only one model is needed at test time.

To improve generalization and avoid co-adaptation of features, we opted for adding dropout (Srivastava et al. 2014) between all the layers of the network. We experimented with different dropout rates (0.2, 0.4) for the inputs and (0.3, 0.5, 0.7) for the hidden layers obtaining better results with the highest values, i.e., 0.4 and 0.7.

Dataset: Table 1 reports the labels distributions on the train dataset. It is important to note that the dataset for Task C presents a higher number of negative than positive examples. For this reason, we automatically extended the training dataset (ED) with new positive matches for tasks B and C, respectively. This process is done by creating the (q_{rel}, c_{rel}) pairs for each q_{rel} from the training set for Task A and creating triples of the form $(q_{rel}, q_{rel}, c_{rel})$, where the label for question-question similarity is obviously positive and the labels for Task C are inherited from those of Task A. The resulting dataset contains $34,100$ triples and its relevance label distribution is presented in the last row of Table 1. The extended version of the dataset with the annotation for MTL is made available for download for comparison purposes [1].

1 https://ikernels-portal.disi.unitn.it/repository/

Table 2
Impact of CNN vs. LSTM sentence models on the baseline network for Task C.

Model	MAP	MRR
LSTM	43.91	49.28
CNN	44.43	49.01
CNN Train	44.43	49.01
CNN Train + ED[2]	**44.77**	**52.07**

Table 3
Results on the validation and test set for the proposed models

Models	Task A: question-comment similarity				Task B: question-question similarity				Task C: new question-comment similarity			
	DEV		TEST		DEV		TEST		DEV		TEST	
	MAP	MRR	MAP	MRR	MAP	MRR	MAP	MRR	MAP	MRR	MAP	MRR
Random	-	-	59.53	67.83	-	-	46.98	50.96	-	-	15.01	15.19
IR Baseline	-	-	52.80	58.71	-	-	74.75	83.79	-	-	40.36	45.83
Kelp	-	-	79.19	86.42	-	-	-	-	-	-	-	-
UH-PRHLT	-	-	-	-	-	-	76.70	83.02	-	-	-	-
SUper-team	-	-	-	-	-	-	-	-	-	-	55.41	61.48
$\langle q_{rel}, c_{rel} \rangle$	68.93	76.46	74.73	81.18	-	-	-	-	-	-	-	-
$\langle q_{new}, q_{rel} \rangle$	-	-	-	-	74.19	**83.26**	**73.70**	**82.13**	-	-	-	-
$\langle q_{new}, c_{rel} \rangle$	-	-	-	-	-	-	-	-	44.77	52.07	41.95	47.21
$\langle q_{new}, q_{rel}, c_{rel} \rangle$	-	-	-	-	-	-	-	-	45.59	51.04	46.99	55.64
$\langle q_{new}, q_{rel}, c_{rel} \rangle + \leftrightarrow$	70.69	77.19	**75.52**	82.11	72.92	80.20	72.88	80.58	47.82	53.03	46.45	51.72
MTL (BC)	-	-	-	-	**74.22**	80.40	73.68	81.59	47.80	52.31	48.58	**55.77**
MTL (AC)	70.11	76.50	75.43	**82.46**	-	-	-	-	46.34	51.54	48.49	54.01
MTL (ABC)	69.93	76.27	74.42	81.68	70.68	75.85	71.07	80.11	**49.63**	**55.47**	49.87	55.73
MTL (ABC)*	**70.70**	**77.48**	74.89	81.80	74.21	81.93	72.23	80.33	**49.63**	**55.47**	49.87	55.73
MTL (weighted score)	-	-	-	-	-	-	-	-	-	-	**52.67**	55.68

Measures: we report the results of our models in terms of MAP and MRR. Both provide a higher score if the relevant items are higher in the rank. However, MAP takes into account the rank of all of the relevant items with respect to the irrelevant ones. MRR only considers the first relevant retrieved item with respect to all the others.

5.2 Impact of the sentence models

Table 2 shows a comparison between CNN and LSTM sentence models when used in our general architecture (see Sec. 3) for solving Task C. We derived the results from the development set [3]. We observe that the two sentence models show comparable results. For the rest of the experiments, we used the CNN sentence model, since it shows faster convergence rate and more stable results with respect to the LSTM sentence model. In the second part of Table 2, we demonstrate that using the extended dataset for solving Task C leads to higher results than the original one. In particular, we noted that there is an improvement of 3 points in MRR.

2 Extended Dataset for Task C computed using questions from Task A.
3 In this work, the dataset Train-part2 were used as development set.

5.3 Results of individual models

Table 3 shows the results of our individual and MTL models, in comparison with the Random and Information Retrieval baselines of the challenge (first grouped row), and the three-top systems of SemEval 2016, KeLP (Filice et al. 2016), UH-PRHLT (Franco-Salvador et al. 2016), SUper-team (Mihaylova et al. 2016) (second grouped row).

The third grouped row shows the performance of the individual models when trained on input pairs, $\langle q_{rel}, c_{rel} \rangle$, $\langle q_{new}, q_{rel} \rangle$ and $\langle q_{new}, c_{rel} \rangle$ for task A, B and C, respectively. The model for the three tasks is the same (described in Sec. 3). These results show that the individual models can generalize well enough on all tasks. In particular, on Task B, they achieve the best results of our proposed model (the numbers in bold indicate the best results among the proposed models).

The fourth grouped row illustrates the models exploiting the joint input, $\langle q_{new}, q_{rel}, c_{rel} \rangle$, but no joint learning is carried out, i.e., the networks for the different tasks are trained individually. The results show that a small degradation of performance happens in Task B, while Task A slightly improves. These variations may be due to the fact that tasks A and B can be efficiently solved using the standard pairwise approach, thus the extra text introduced in the model may just add some noise. However, using the shared sentence model for q_{new} and q_{rel} of the tasks B and C (indicated with $\leftrightarrow$) improves the overall performance.

5.4 Results of MTL models

The shared input representation shows good results on all tasks, thus, in the last set of experiments, we jointly trained (i) tasks B and C, (ii) tasks A and C and finally (iii) the three tasks together.

The results are reported in the fifth grouped row. It is interesting to note that the major boost in terms of performance is obtained when we jointly train all the three tasks. In fact, the MTL architecture improves the individual model in terms of MAP by about 2 absolute points on the DEV set and by 3 absolute points on the TEST set for Task C, while the performance on the other tasks tends to degrade. However, if the three different models are evaluated at different epochs of training, e.g., see MTL(ABC)*, it is possible to obtain accuracy comparable to the individual models for all the three tasks. As previously explained, when applying MTL, the individual objective functions converge at different epochs. Therefore, when the global loss reaches the minimum, it is possible that individual models are sub-optimal.

Indeed, the comparison between the learning curves (on the development set) for Task B (Figure 4) and Task C (Figure 5) shows that for the former, models achieve earlier convergence rate (epoch 2) while for the latter they converge later (epoch 16). Moreover, Figure 3 shows that the results on Task A are not badly affected by jointly training models with the other two tasks.

Finally, the learning curves show that our networks trained in MTL tend to have faster convergence rate than the individual models: this is a very interesting result.

We also experimented with shallower networks and SVMs using the prediction scores from the different classifiers in a stacking approach, and obtained results far below the baselines[4].

4 We did not include these results as they do not provide interesting findings.

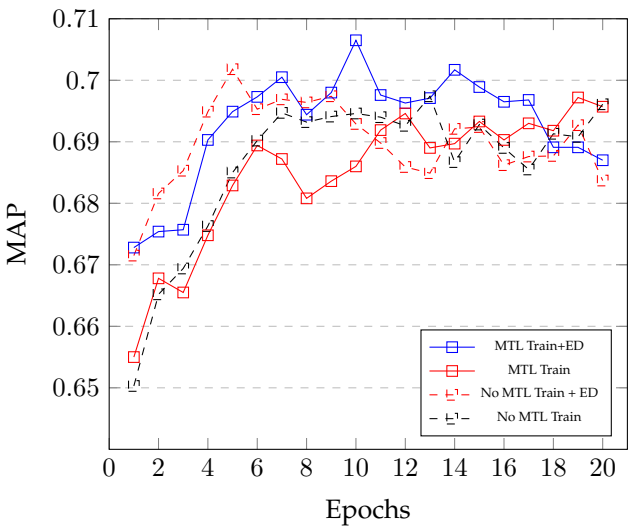


Figure 3
Learning curves for Task A on the dev. set; dotted and solid lines represent the individual and multi-task models, respectively.

Comparison with the state of the art. Our models would have ranked 4^{th} on Task C of the Semeval 2016 competition [5], i.e., the main task of the challenge. In contrast, our models for the other two tasks, which do not benefit from the overall MTL architecture would have achieved a middle position (8^{th}). These results are important since our proposed MTL architecture obtains a placement very close to the top system, without requiring task-specific features, which in cQA are extremely important, e.g., the thread-level features.

Finally, one reason of why we do not achieve the state of the art on Task C is due to the difference between training and test data. Several challenge participants solved this problem by using a weighted sum between the score of the Task A classifier and the Google rank as a strong features for modeling Task C. We followed a similar approach estimating the weight MTL on the dev set and using the computed score to rank the comments of the test set. This improved the MAP of our MTL by about 2.8 absolute points on the test set, obtaining a result comparable with the model ranked 2^{nd} on Task C at the Semeval 2016 competition.

6. Related Work

Previous work related to the topics presented in this paper spans three major research areas: Question Retrieval (targeting question similarity), Passage Reranking (targeting question and answer similarity) and MTL. In the following, we will report the most important works in these areas.

5 http://alt.qcri.org/semeval2016/task3/index.php?id=results

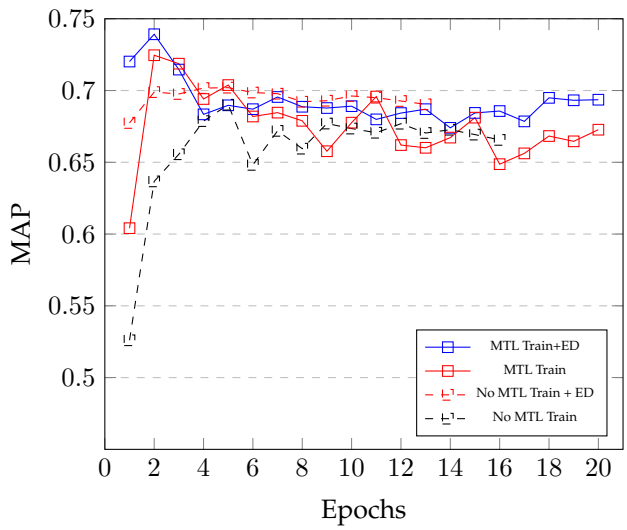


Figure 4
Learning curves for Task B on the development set; dotted lines represent the individual models, while the solid lines represent the multi-task ones.

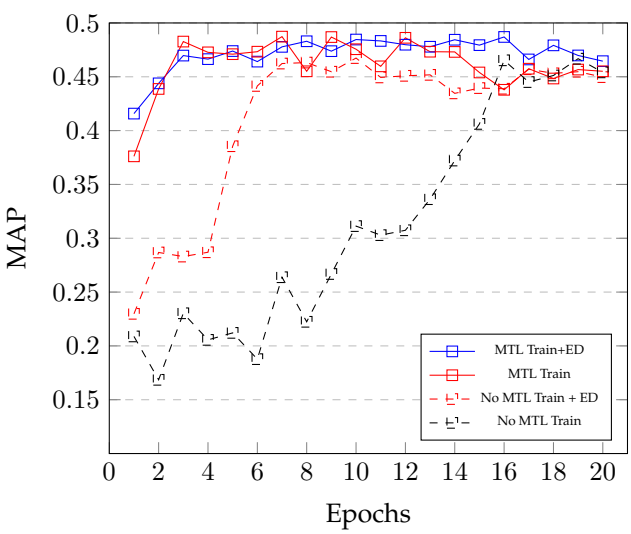


Figure 5
Learning curves for Task C on the development set; dotted lines represent the individual models, while the solid lines represent the multi-task ones.

Question-Question Similarity. Determining question similarity remains one of the main tasks needed to be solved in cQA due to difficult problems such as "lexical gap". Early approaches on question similarity used statistical machine translation techniques to measure similarity between questions. For example, [Jeon, Croft, and Lee 2005] and [Zhou et al. 2011] used a language models based on word or phrase translation probabilities to estimate similarity between questions. However, effective approaches based

on statistical machine translation require lots of data to estimate word probabilities. Language models for question-question similarity were also explored by [Cao et al. 2009]. These models exploit information from the category structure of Yahoo! Answers when computing similarity between two questions. Instead, [Duan et al. 2008] propose an approach that identifies the topic and focus in a text and compute similarity between two input questions by matching the extracted topic and focus information. A different approach to question-question similarity is provided by [Ji et al. 2012] and [Zhang et al. 2014]. They use LDA to learn the probability distribution over the topics that generate the question/answers pairs. Later, this distribution is used to measure similarity between questions.

Question-Answer Similarity. In recent years, many models have been proposed for computing similarity of an answer with respect to a question. For example, [Yao et al. 2013] trained a conditional random field based on a set of powerful features, such as tree-edit distance between question and answer trees: these also enable the extraction of answers from pre-retrieved sentences. [Heilman and Smith 2010] use a linear classifier using syntactic features to solve different tasks such as recognizing textual entailment, paraphrases and answer selection. [Wang, Smith, and Mitamura 2007] propose the use of Quasi-synchronous grammars to select short answers for TREC questions. This is done by learning syntactic and semantic transformation from the question to the answer trees. [Wang and Manning 2010] propose a probabilistic Tree-Edit model with structured latent variables for solving textual entailment and question answering. An advanced model based on structural representations was proposed in (Moschitti et al. 2007; Moschitti 2008; Severyn and Moschitti 2012; Severyn, Nicosia, and Moschitti 2013; Severyn and Moschitti 2013, 2015; Tymoshenko and Moschitti 2015). These model use SVM with kernels to learn structural patterns between questions and answers encoded in form of shallow syntactic parse trees.

Finally, [Wang and Nyberg 2015] trained a long short-term memory model for selecting answers to TREC questions. Their model takes words from question and answer sentences as input and returns a score measuring the relevancy of an answer with respect to a given question. A recent work close to ours is (Guzmán, Màrquez, and Nakov 2016), where the authors build a neural network for solving Task A of SemEval. However, this does not approach the problem as MTL.

Related work on MTL. A good overview on MTL, i.e., learning to solve multiple tasks by using a shared representation with mutual benefit, is given in (Caruana 1997). [Collobert and Weston 2008] trained a convolutional NN with MTL which, given an input sentence, performs many sequence labeling tasks. They showed that jointly training their system on different tasks, such as speech tagging, named entity recognition, etc., significantly improves the performance on the main task, i.e., semantic role labeling, without requiring hand-engineered features.

[Liu et al. 2015] is the most close work to ours. They used multi-task deep neural networks to map queries and documents into semantic vector representations. This representation is later used into two tasks: query classification and question-answer reranking. The results showed a competitive gain over strong baselines. In our work, we have presented an architecture that can also exploit joint representation of question and comments, given the strong interdependencies among the different SemEval Tasks.

7. Conclusion

In this paper we proposed several Deep Neural Networks for the task of automatic cQA. Our main result is a network that can effectively exploit the characteristics of the cQA task to carry out interesting MTL. Our network designed and trained in an MTL setting shows better accuracy and a higher convergence rate than the models independently trained. The results show that our MTL model approaches the performance of the models participating at the SemEval 2016 cQA competition. It should be noted that all the other challenge systems use domain specific features, which are both very important but also rather costly to engineer.

In the future, we would like to use more effective features and combine them with other machine learning methods.

References

Cao, Xin, Gao Cong, Bin Cui, Christian Søndergaard Jensen, and Ce Zhang. 2009. The use of categorization information in language models for question retrieval. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 265–274, New York, NY, USA. ACM.

Caruana, Rich. 1997. Multitask learning. *Machine Learning*, 28(1):41–75, Jul.

Collobert, Ronan and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 160–167, New York, NY, USA. ACM.

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

Duan, Huizhong, Yunbo Cao, Chin-Yew Lin, and Yong Yu. 2008. Searching questions by identifying question topic and question focus. In *Proceedings of ACL-08: HLT*, pages 156–164. Association for Computational Linguistics.

Filice, Simone, Danilo Croce, Alessandro Moschitti, and Roberto Basili. 2016. Kelp at semeval-2016 task 3: Learning semantic relations between questions and answers. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1116–1123. Association for Computational Linguistics.

Franco-Salvador, Marc, Sudipta Kar, Thamar Solorio, and Paolo Rosso. 2016. Uh-prhlt at semeval-2016 task 3: Combining lexical and semantic-based features for community question answering. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 814–821. Association for Computational Linguistics.

Guzmán, Francisco, Lluís Màrquez, and Preslav Nakov. 2016. Machine translation evaluation meets community question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 460–466. Association for Computational Linguistics.

Heilman, Michael and Noah A. Smith. 2010. Tree edit models for recognizing textual entailments, paraphrases, and answers to questions. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 1011–1019. Association for Computational Linguistics.

Jeon, Jiwoon, W. Bruce Croft, and Joon Ho Lee. 2005. Finding similar questions in large question and answer archives. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, CIKM.

Ji, Zongcheng, Fei Xu, Bin Wang, and Ben He. 2012. Question-answer topic model for question retrieval in community question answering. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 2471–2474, New York, NY, USA. ACM.

Kalchbrenner, Nal, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 655–665. Association for Computational Linguistics.

Kim, Yoon. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751. Association for Computational Linguistics.

Liu, Xiaodong, Jianfeng Gao, Xiaodong He, Li Deng, Kevin Duh, and Ye-Yi Wang. 2015. Representation learning using multi-task deep neural networks for semantic classification and information retrieval. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 912–921. Association for Computational Linguistics.

Mihaylova, Tsvetomila, Pepa Gencheva, Martin Boyanov, Ivana Yovcheva, Todor Mihaylov, Momchil Hardalov, Yasen Kiprov, Daniel Balchev, Ivan Koychev, Preslav Nakov, Ivelina Nikolova, and Galia Angelova. 2016. Super team at semeval-2016 task 3: Building a feature-rich system for community question answering. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 836–843. Association for Computational Linguistics.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In Chris J.C. Burges, Léon Bottou, Max Welling, Zoubin Ghahramani, and ilian.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Moschitti, Alessandro. 2008. Kernel methods, syntax and semantics for relational text categorization. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM 2008*, pages 253–262, Napa Valley, California, USA, October 26-30.

Moschitti, Alessandro, Silvia Quarteroni, Roberto Basili, and Suresh Manandhar. 2007. Exploiting syntactic and shallow semantic kernels for question answer classification. In *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic*.

Prechelt, Lutz. 1998. Early stopping-but when? In *Neural Networks: Tricks of the trade*. Springer, pages 55–69.

Severyn, Aliaksei and Alessandro Moschitti. 2012. Structural relationships for large-scale learning of answer re-ranking. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 741–750. ACM.

Severyn, Aliaksei and Alessandro Moschitti. 2013. Automatic feature engineering for answer selection and extraction. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, EMNLP '13, pages 458–467, Seattle, Washington, USA.

Severyn, Aliaksei and Alessandro Moschitti. 2015. Learning to rank short text pairs with convolutional deep neural networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 373–382. ACM.

Severyn, Aliaksei and Alessandro Moschitti. 2016. Modeling relational information in question-answer pairs with convolutional neural networks. *CoRR*, abs/1604.01178.

Severyn, Aliaksei, Massimo Nicosia, and Alessandro Moschitti. 2013. Building structures from classifiers for passage reranking. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 969–978. ACM.

Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Tieleman, Tijmen and Geoffrey Hinton. 2012. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning*, 4.

Tymoshenko, Kateryna and Alessandro Moschitti. 2015. Assessing the impact of syntactic and semantic structures for answer passages reranking. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM 2015*, pages 1451–1460, Melbourne, VIC, Australia, October 19 - 23.

Wang, Di and Eric Nyberg. 2015. A long short-term memory model for answer sentence selection in question answering. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 707–712. Association for Computational Linguistics.

Wang, Mengqiu and Christopher Manning. 2010. Probabilistic tree-edit models with structured latent variables for textual entailment and question answering. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1164–1172. Coling 2010 Organizing Committee.

Wang, Mengqiu, Noah A. Smith, and Teruko Mitamura. 2007. What is the jeopardy model? a quasi-synchronous grammar for qa. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*.

Yao, Xuchen, Benjamin Van Durme, Chris Callison-Burch, and Peter Clark. 2013. Answer extraction as sequence tagging with tree edit distance. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 858–867. Association for Computational Linguistics.

Zhang, Kai, Wei Wu, Haocheng Wu, Zhoujun Li, and Ming Zhou. 2014. Question retrieval with high quality answers in community question answering. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, CIKM '14, pages 371–380, New York, NY, USA. ACM.

Zhou, Guangyou, Li Cai, Jun Zhao, and Kang Liu. 2011. Phrase-based translation model for question retrieval in community question answer archives. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 653–662. Association for Computational Linguistics.