ISSN 2499-4553

IJCoL

Italian Journal of Computational Linguistics

Rivista Italiana di Linguistica Computazionale

Volume 5, Number 2
december 2019

aA Accademia 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 979-12-80136-06-0

Accademia University Press
via Carlo Alberto 55
I-10123 Torino
info@aAccademia.it
www.aAccademia.it/IJCoL_5_2

Accademia University Press è un marchio registrato di proprietà
di LEXIS Compagnia Editoriale in Torino srl

IJCoL

Volume 5, Number 2
december 2019

CONTENTS

Nota editoriale

Roberto Basili*
Università di Roma, Tor Vergata

Simonetta Montemagni**
ILC - CNR

Il secondo numero del quinto anno della rivista *Italian Journal of Computational Linguistics* (*IJCoL*), la rivista italiana promossa dall'*Associazione Italiana di Linguistica Computazionale* (AILC - `www.ai-lc.it`), è un volume miscellaneo i cui articoli documentano una selezione di linee di ricerca attive nel panorama della Linguistica Computazionale italiana con risultati interessanti. Tra questi, vi sono articoli che documentano lavori di ricerca risultati particolarmente promettenti nell'ambito della Conferenza CLiC-it 2019 (Bari, 13–15 novembre 2019), così come contributi originali proposti per la pubblicazione sulla rivista. Tutti i contributi sono stati sottoposti a un processo di peer-review, iterativo nel caso degli articoli premiati come "Best Young Paper" e "Distinguished Young Paper" nell'ambito della conferenza. Chiude il volume un contributo dedicato alla memoria di Emanuele Pianta, un ricercatore che ha significativamente contribuito alla crescita della Linguistica Computazionale in Italia scomparso prematuramente nel 2012.

I temi affrontati coprono sviluppi recenti e fecondi della ricerca in linguistica computazionale, come ad esempio l'uso di tecniche di NLP complesse per la analisi dei fenomeni legati ai social media o l'adozione di algoritmi neurali per il trattamento di fenomeni audio (*speech profiles*) o visuali (video e immagini) e l'ottimizzazizone di compiti linguistici complessi, rappresentati dal cosiddetto "*captioning*" o "*speech recognition*".

Il lavoro di Polignano *e colleghi* presenta AlBERTo, un modello di lessico semantico per la lingua italiana addestrato sulla lingua dei Social Media, in particolare Twitter. In linea con i modelli basati sul paradigma dei *trasformers* (BERT *in primis*), AlBERTo è stato addestrato sfruttando la decomposizione del task di apprendimento nell'ambiente Google Cloud Platform e la disponiblità del corpus TWITA che raccoglie circa 200 milioni di tweet generalisti in lingua italiana. Il modello risultante è distribuito *open source* attraverso la piattaforma GitHub. La disponibilità di tale risorsa su larga scala è un risultato importante, in quanto rende possibili numerose ricerche e applicazioni di "*Computational Social Science*" per l'italiano, da parte di una sempre più vasta comunità di ricercatori.

Il lavoro di Ferro e Tamburini valida ed estende il ruolo di modelli di *smoothing*, discusso recentemente, negli algoritmi di *Pitch Detection* impiegati in sistemi di *Speech Recognition*. La ricerca dimostra che gli algoritmi neurali per lo *smoothing* possono migliorare le performances in modo significativo. In particolare, viene introdotto un *pitch smoother* basato su una architettura neurale che usa Keras come interfaccia di riferimento verso TensorFlow. Esso è in grado di incidere in modo eccellente su due *benchmark* standard per la lingua inglese, apprendendo il meccanismo di smoothing di un *pitch detector* in modo da eliminare completamente alcune classi di errori.

* Dept. of Enterprise Engineering - Via del Politecnico 1, 00133 Roma
E-mail: `basili@info.uniroma2.it`
** Istituto di Linguistica Computazionale "A. Zampolli", CNR - Via Moruzzi 1, 56124 Pisa
E-mail: `simonetta.montemagni@ilc.cnr.it`

Il lavoro di Scaiella *e colleghi* presenta l'applicazione di tecniche neurali per l'addestramento di un sistema di generazione di commenti testuali a immagini e testi. La ricerca sfrutta architetture in grado di codificare video (o immagini) in vettori numerici (*embeddings*) per alimentare un secondo sistema neurale (ricorrente) usato per generare il commento in lnguaggio naturale. Il lavoro descrive lo sviluppo semiautomatico di un corpus di video commentati per la lingua italiana, usando come sorgente la controparte in inglese.

Il lavoro di Monti e Di Buono descrive una risorsa originale e innovativa, il corpus PARSEME-It VMWE sviluppato all'interno della PARSEME COST Action che rappresenta il primo e l'unico corpus per la lingua italiana ad oggi arricchito con informazione relativa a una vasta a variegata tipologia di espressioni polirematiche (*MultiWord Expressions*, in breve MWE), che vanno da espressioni idiomatiche e composti a *light verb contructions* e locuzioni di varia natura (avverbiali, preposizionali, etc.). Il corpus PARSEME-It VMWE italiano rappresenta l'esito di un'analisi estensiva e linguisticamente motivata delle MWEs italiane, ed è accompagnato da specifiche dettagliate per la loro identificazione, classificazione e rappresentazione.

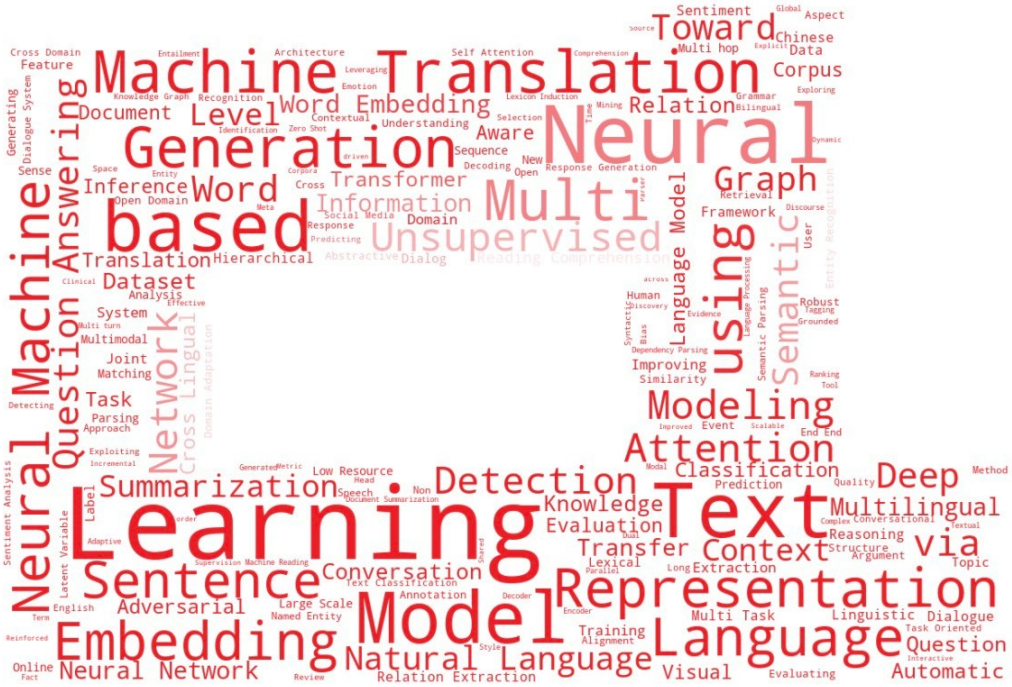
Infine, segue il contributo dedicato a Emanuele Pianta, eccellente studioso e ricercatore del settore della Linguistica Computazionale e in particolare del Trattamento Automatico della Lingua, prematuramente scomparso nel Novembre 2012. Magnini, Delmonte e Tonelli - tra i ricercatori che sono stati più vicini a Emanuele - ripercorrono i suoi contributi alla ricerca, che vengono presentati come un esempio vivido e fecondo per molti ricercatori, giovani e meno giovani. In riconoscimento del suo contributo, il Direttivo di AILC ha deciso di attivare un Premio intitolato alla sua memoria, assegnato annualmente alla miglior tesi di laurea magistrale nell'ambito della Linguistica Computazionale discussa in una università italiana.

Speriamo che questa sintesi del volume - inevitabilmente parziale - ispiri, come sempre, al lettore il desiderio di navigare, secondo i propri interessi, nelle pieghe delle pagine di questo volume, certamente più ricche di dettagli e sfumature.

Prima di chiudere questa nota vogliamo segnalare un importante evento che ha visto il coinvolgimento della comunità italiana della Linguistica Computazionale: il 57° Convegno Annuale dell'*Association for Computational Linguistics* (ACL), la più importante associazione scientifica internazionale per la Linguistica Computazionale, che si è svolto alla Fortezza da Basso a Firenze dal 28 luglio al 2 agosto 2019. Il convegno annuale dell'ACL è il momento in cui scienziati di tutto il mondo si confrontano per fare il punto sullo stato dell'arte della disciplina e sulle prospettive future di sviluppo. L'edizione italiana del 2019 è stata eccezionale per due motivi.

Prima di tutto è la prima volta che il convegno di ACL è stato organizzato in Italia. Organizzatori locali dell'evento sono stati Alessandro Lenci (Università di Pisa), Bernardo Magnini (Fondazione Bruno Kessler, Trento), Simonetta Montemagni (Istituto di Linguistica Computazionale "A. Zampolli" del CNR), che si sono avvalsi della collaborazione di un ampio segmento della comunità italiana in questo settore. Il fatto che ACL abbia scelto l'Italia come paese ospite del suo convengo annuale è stato un grande onore per tutta la nostra comunità e può essere visto come testimonianza della sua rilevanza nel panorama internazionale. L'Italia è in effetti da sempre protagonista delle ricerche in linguistica computazionale, che ha mosso i suoi primi passi proprio a Pisa ormai più di 50 anni fa. Oggi l'Italia conta molti centri di ricerca e ditte che contribuiscono attivamente all'avanzamento dello stato dell'arte nel settore. La ricchezza di attività della comunità nazionale è testimoniata dalla recente nascita dell'*Associazione Italiana di Linguistica Computazionale* (AILC), che ha supportato attivamente l'organizzazione del convegno.

Il secondo fattore di eccezionalità del convegno fiorentino sono state le sue dimensioni. Tutti i numeri di ACL 2019 sono da record: 660 presentazioni accettate alla conferenza principale in 6 sessioni parallele, 21 tra workshop e conferenze satellite per più di 3200 iscritti in totale. L'edizione 2019 è in assoluto quella col maggior numero di partecipanti nella storia di ACL. Tra i temi più presenti nel programma ci sono stati gli agenti conversazionali in grado di dialogare con gli esseri umani, la traduzione automatica, l'uso dei metodi di "deep learning", ma anche aspetti etici legati alle tecnologie del linguaggio. Di grande attualità sono risultate le nuove tecnologie per combattere l'uso improprio e offensivo del linguaggio sui Social Media, o per gestire conversazioni tra medico e paziente, solo per menzionarne alcune. La varietà dei temi affrontati emerge chiaramente dal "word cloud" che segue, che contiene le parole chiave emergenti dall'analisi dei titoli degli articoli presentati nell'ambito della conferenza.



La conferenza è stata sponsorizzata da tutte le "Big Companies" del Web (Amazon, Apple, Baidu, Facebook, Google, ecc.) e da molte altre imprese italiane e straniere, segno del grande interesse commerciale di un settore di mercato in continua espansione.

Qualcuno ha definito la Conferenza *ACL-2019* tenutasi a Firenze "*The best ACL ever!*": ciò può anche essere visto come testimonianza dell'eccellenza del contributo alla ricerca internazionale che la comunità italiana della Linguistica Computazionale fornisce da anni e che AILC intende rafforzare e promuovere sin dai suoi esordi, solo qualche anno fa. La strada percorsa è già lunga, nonostante la giovane età dell'associazione: ne siamo orgogliosi.

1. Editorial Note Summary

The second volume of the fifth year of the *Italian Journal of Computational Linguistics* (IJCoL) promoted by the *Associazione Italiana di Linguistica Computazionale* (AILC - `www.ai-lc.it`) integrates original research results as well as papers that have been presented at CLiC-it 2019 held in Bari, in November 2019. The major themes discussed by the papers are hot topics that include complex learning methods for the analysis of Social Media texts, neural algorithms for integrated audio, visual and language learning. Some works are on NLP resources developed for teh Italian language.

The interetsing paper by Polignano *et al.* presents AlBERTo, a lexical semantic model based on the Transformer paradigm, trained over Social Media material in Italian. AlBERTo exploits the availability of the TWITA corpus, that includes about 200 millions generalist tweets in Italian. The resulting model is distributed under an *open source* scheme on the GitHub platform. The availability of this resource is inspiring a number of further studies on "*Computational Social Science*" over Web sources in Italian involving a growing research community.

Ferro and Tamburini study how neural *smoothing models* can be adopted to improve the *Pitch Detection* stage in *Speech Recognition* systems. The work shows how a *pitch smoother* based on a Keras API towards interfaccia Tenworflow is able to limit error rates of a *pitch detector* on a large English speech corpus.

The work by Scaiella and colleagues presents the application of convolutional and recurrent neural networks to the task of automatic captioning of images an video. The architecture develops on methods already experimented for English, and shows how on Italian similar performances can be achieved. The work also describes the semi-automatic development and the release of an annotated corpus of video captions for the Italian language, through the automatic translation of the English counterpart.

The work by Monti and Di Buono presents the PARSEME-It corpus for the analysis of multiword expressions (MWE). It is a linguistically principled annotated corpus, that embody a comprehensive study for the annotation of MWE in italian: it specializes the PARSEME COST Action framework.

Finally, one contribute to this volume is dedicated to Emanuele Pianta, brilliant researcher in Computational Linguistics and Natural Language Processing, whose untimely death, in November 2012, has left a sad gap in the Italian CL community. Bernardo Magnini, Rodolfo Delmonte and Sara Tonelli, who were closer to Emanuale during his studies, go through his own major research contributions in the paper. It surveys thus a lively and fruitful example for all of us, younger or senior researchers. Accordingly, in an attempt to emphasize his contributions, the Steering Committee of AILC decided to dedicate to Emanuele Pianta a Prize, yearly assigned to the best Master Degree thesis in the Computational Linguistics area defended during the year in one Italian University.

This very synthetic view serves only to survey the focus of the papers. We leave the reader the pleasure to navigate across the valuable pages of our volume and discover there all the interesting details.

AlBERTo: Modeling Italian Social Media Language with BERT

Marco Polignano *
University of Bari A. Moro

Valerio Basile **
University of Turin

Pierpaolo Basile †
University of Bari A. Moro

Marco de Gemmis ‡
University of Bari A. Moro

Giovanni Semeraro §
University of Bari A. Moro

Natural Language Processing tasks recently achieved considerable interest and progresses following the development of numerous innovative artificial intelligence models released in recent years. The increase in available computing power has made possible the application of machine learning approaches on a considerable amount of textual data, demonstrating how they can obtain very encouraging results in challenging NLP tasks by generalizing the properties of natural language directly from the data. Models such as ELMo, GPT/GPT-2, BERT, ERNIE, and RoBERTa have proved to be extremely useful in NLP tasks such as entailment, sentiment analysis, and question answering. The availability of these resources mainly in the English language motivated us towards the realization of AlBERTo, a natural language model based on BERT and trained on the Italian language. We decided to train AlBERTo from scratch on social network language, Twitter in particular, because many of the classic tasks of content analysis are oriented to data extracted from the digital sphere of users. The model was distributed to the community through a repository on GitHub and the Transformers library (Wolf et al. 2019) released by the development group huggingface.co. We have evaluated the validity of the model on the classification tasks of sentiment polarity, irony, subjectivity, and hate speech. The specifications of the model, the code developed for training and fine-tuning, and the instructions for using it in a research project are freely available.

1. Introduction and Motivation

The diffusion of text representation models based on probabilistic approaches has contributed significantly to the adoption of innovative models for understanding natural language. A basic approach, simply based on frequencies, already has the capacity of generalizing on the text to represent a document as a set of numerical vectors, one for each term contained in it. However, such strategy does not address several problems related with this representation, such as the possibility that a common term is not

* Dept. of Computer Science, Via E.Orabona 4, Bari, Italy. Email: marco.polignano@uniba.it
** Dept. of Computer Science, Corso Svizzera 185, Turin, Italy. Email: valerio.basile@unito.it
† Dept. of Computer Science, Via E.Orabona 4, Bari, Italy. Email: pierpaolo.basile@uniba.it
‡ Dept. of Computer Science, Via E.Orabona 4, Bari, Italy. Email: marco.degemmis@uniba.it
§ Dept. of Computer Science, Via E.Orabona 4, Bari, Italy. Email: giovanni.semeraro@uniba.it

always a good indicator of the content of the document, or the absence of focus on the word order. Starting from these fundamental problems, scientific research has moved towards increasing the complexity of numerical representations of text such as TF-IDF, Latent Semantic Indexing, Random Indexing, and Page-Rank, to name but a few. Using such implementation strategies, numerous NLP tasks, including machine translation, text classification, and question answering, have obtained a remarkable improvement in terms of performance and reliability. For instance, consider the effectiveness of Google Translate in the 2000s and the quality of its translation in recent years. In particular, a significant contribution was made by the advent of distributional semantics models such as word embedding.

Mikolov et al. (2013) notably contributed to the genesis of numerous strategies for representing terms based on the idea that semantically related terms have similar vector representations. They showed exciting arithmetic properties of their vector representation, such as the sum of two terms returning a new semantically consistent vector that is equivalent to the linguistic sum of them. The famous representation "King - Man + Woman $\sim$ Queen" is a teaching example. Such approaches as Word2Vec (Mikolov et al. 2013), Glove (Pennington, Socher, and Manning 2014), and FastText (Bojanowski et al. 2017) suffer from the problem that multiple concepts, associated with the same term, are not represented by different word embedding vectors in the distributional space (the representation is *context-free*). This means that each term has only a single word embedding representation in the distributional space, and different concepts of the same term are not represented. Moreover, it has been demonstrated that they do not perform well when applied to different domains from the one on which they have been learned (Polignano et al. 2019a).

New strategies such as ELMo (Peters et al. 2018), GPT/GPT-2 (Solaiman et al. 2019), and BERT (Devlin et al. 2019) overcome this limit by learning a language model for a contextual and task-independent representation of terms. In particular, these models are trained to predict the totality or a span of the starting sentence. This allows them to compute a model able to predict the most probable word from its vocabulary in a specific context (often both previous and subsequent). Recently, several articles have demonstrated the effectiveness of this technique in almost all NLP tasks in the English language, and recently, multilingual models have been distributed. In their multilingual version, they mainly use a mix of text obtained from large corpora in different languages to build a general language model to be reused for every application in any language. As reported by the BERT documentation "the Multilingual model is somewhat worse than a single-language model. However, it is not feasible for us to train and maintain dozens of single-language model." This entails significant limitations related to the type of language learned (concerning the document style) and the size of the vocabulary.

These reasons have led us to create the equivalent of the BERT model for the Italian language and specifically on the language style used on Twitter: **AlBERTo**. This idea was supported by the intuition that many NLP tasks for the Italian language are carried out for the analysis of social media data, both in business and research contexts. In this paper, we present AlBERTo, providing the details of its architecture and training procedure. We furthermore present the results of experiments showing that AlBERTo significantly improves over the state of the art in sentiment analysis and hate speech detection benchmarks in the Italian language.

The present article is based on, and extends, the work reported in Polignano et al. (2019c) and Polignano et al. (2019b).

2. Background and Related Work

A Task-Independent Language Model is based on the idea of creating a deep learning architecture, particularly an encoder and a decoder, so that the encoding level can be used in more than one NLP task. In this way, it is possible to obtain a decoding level with weights optimized for the specific task (fine-tuning). A general-purpose encoder should therefore be able to provide an efficient representation of the terms, their position in the sentence, context, grammatical structure of the sentence, semantics of the terms. The idea behind such models is that if a model can predict the next word that follows in a sentence, then it is able to generalize the syntactic and semantic rules of the language.

One of the first systems able to satisfy these requirements was ELMo (Peters et al. 2018), based on a large BiLSTM neural network (2 BiLSTM layers with 4,096 units and 512 dimension projections and a residual connection from the first to the second layer) trained for 10 epochs on the 1B WordBenchmark (Chelba et al. 2014). The goal of the network was to predict the same starting sentence in the same initial language (like an autoencoder). It has proved the correct management of polysemy by demonstrating its efficacy on six different NLP tasks for which it obtained state-of-the-art results: Question Answering, Textual Entailment, Semantic Role labeling, Coreference Resolution, Name Entity Extraction, and Sentiment Analysis.

Following the basic idea of ELMo, another language model called GPT has been developed in order to improve the performance of the tasks included in the GLUE benchmark (Wang et al. 2018). GPT replaces the BiLSTM network with a Transformer architecture (Vaswani et al. 2017). A Transformer is an encoder-decoder architecture that is mainly based on feed-forward and multi-head attention layers. Moreover, in Transformers, terms are provided as input without a specific order. Consequently, a positional vector is added to the term embeddings in order to encode the information which comes from the position of the term into the sentence. Unlike ELMo, in GPT, for each new task, the weights of all levels of the network are optimized, and the complexity of the network (in terms of parameters) remains almost constant. Moreover, during the learning phase, the network does not limit itself to a single sentence but it splits the text into spans to improve the predictive capacity and the generalization power of the network. The deep neural network is a 12-layer decoder-only transformer with masked self-attention heads (768 dimensional states and 12 attention heads) trained for 100 epochs on the BooksCorpus dataset (Zhu et al. 2015). This strategy proved to be successful compared to the results obtained by ELMo on the same NLP tasks.

BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al. 2019) was developed to work with a strategy very similar to GPT. In its basic version, it is also trained on a Transformer network with 12 encoding levels, 768 dimensional states and 12 heads of attention for a total of 110M of parameters trained on BooksCorpus (Zhu et al. 2015) and Wikipedia English for 1M of steps. The main difference is that the learning phase is performed by scanning the span of text in both directions, from left to right and from right to left, as was already done in BiLSTMs. Moreover, BERT uses a "masked language model": during the training, random terms are masked in order to be predicted by the net. Jointly, the network is also designed to potentially learn the next span of text from the one given in input. These variations on the GPT model allow BERT to be the current state of the art language understanding model. Larger versions of BERT (BERT large) and GPT (GPT-2) have been released and are scoring better results than the normal scale models but they require much more computational power. The base BERT model for the English language is precisely the same used for learning the Italian Language Model (AlBERTo).

Since the release of BERT, several alternative versions have been released. In particular, DistilBERT (Sanh et al. 2019) aims to reduce the time needed to train the model paying it with a minimal loss in performance. DistilBERT uses half the number of BERT learning parameters and retains 97% of its performance. It uses the distilling technique (Buciluǎ, Caruana, and Niculescu-Mizil 2006) that opts for an approximation of very large networks with a network of a much smaller size using the technique of a posteriori approximation. On the contrary, RoBERTa (Liu et al. 2019) proposes itself as a technique to improve the accuracy of the BERT model by training its network on a more significant amount of data. In particular, RoBERTa is trained on 160 GB of text data against the 16 GB used by BERT for about five times longer training time. To make the training phase faster, in RoBERTa the learning strategy "Next Sentence to Predict" has been removed, as well as in AlBERTo, replacing it with a masking learning strategy in which the word hidden during training varies at each time. ERNIE 2.0(Sun et al. 2020) was developed with the aim of improving BERT's learning strategy. In particular, it is based on a multitask learning strategy in order to learn much more information about the vocabulary, syntax, and semantics shared between the different tasks and, therefore, intrinsic in natural language. ERNIE's approach is also incremental, allowing it to accumulate knowledge as the tasks grafted into the model and the training phases performed grow. The ERNIE model is currently better performing than BERT, RoBERTa, ELMo, and GPT, on the GLUE benchmark platform (Wang et al. 2018).

3. AlBERTo

Language resources available on languages other than English are often difficult to find, often leaving local communities in a difficult situation when they need to carry out NLP operations in their own language. The Italian computational linguistics community, on the contrary, manages to be very active in the field and to make available numerous resources often on a par with those available for international languages[1]. Considering the international focus on language models generated through deep neural networks and the absence of them in Italian, it was decided to contribute to the availability of Italian language resources by training a BERT model (Devlin et al. 2019) for Italian from scratch (AlBERTo). This process was divided into two phases. The first included the need to develop code that could be integrated with the one released by Google[2] for BERT. Moreover, it should be reproducible on a virtual machine equipped with a TPU (Tensor processing unit). The use of this technology allows us to work in parallel on a different batch on tensor data. This strategy is indispensable for training models that require very long training time on GPU, such as the 11 days necessarily for BERT base and 22 days for BERT large. In this regard, free credit provided by Google Cloud Platform[3] was used to store the necessary training data on Google Storage Bucket, and instantiate a version of Google Colab[4] Python development environment, on a virtual machine with 25 GB of ram and an 8-core TPU-V2. The second step for training AlBERTo from scratch was to find an Italian language dataset large enough to obtain a model that could accurately generalize its linguistic properties. The choice fell on *TWITA* (Basile, Lai, and Sanguinetti 2018) a collection of domain-generic tweets in Italian extracted

1 For instance, Italian is one of the best represented language in the Universal Dependencies project: https://universaldependencies.org/
2 https://github.com/google-research/bert
3 https://cloud.google.com/
4 https://colab.research.google.com

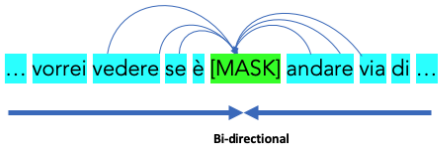


Figure 1
Masking Learning Strategy of BERT and AlBERTo

through API streams and freely usable for research purposes. This dataset meets two requirements that we set as prerequisites. First, the size of the dataset is large enough for a proper training in order to obtain a reliable model. Secondly, it includes a wide range of types of uses of the language. As commonly known, the writing style of social networks is often very different from that used in the common language due to the presence of hashtags, mentions, and contracted words. At the same time, since Twitter contains very heterogeneous tweets, it also includes the use of the Italian language similar to the common one as it is used in official communications, news articles, and advertising messages. It is also very common that text analysis tasks are performed on content extracted from social media, making AlBERTo extremely useful in such contexts. The variety of use of Tweets and the versatility of the resulting model has, therefore, convinced us to use this dataset for the realization of AlBERTo. Consequently, AlBERTo aims to be the first Italian language model to represent the social media language, Twitter in particular, written in the Italian language.

3.1 Model training strategy

The BERT training strategy can be classified as an autoencoder (AE), i.e., unlike an autoregressive strategy (AR), it does not calculate an explicit probability density of a collection of texts but is based on the reconstruction of the suitably perturbed output. This property makes BERT different from other approaches like XLNet (Yang et al. 2019), which, on the contrary, is an AR model. Fig.2 shows the BERT/AlBERTo strategy of learning. The "masked learning" is applied on a $12x$ Transformer Encoder, where, for each input, a percentage of terms is hidden using the [MASK] token and then trained for guessing it in order to optimize network weights in back-propagation (Devlin et al. 2019). Due to the lack of probability estimation of terms in the collection, BERT uses context words to reconstruct the original text portion. Precisely, the context is not calculated as in LSTMs one side at a time but simultaneously on both sides. An example is shown in Fig. 1, which shows that the probability estimation of the hidden word is calculated from the co-occurrences of the context terms. On the one hand, this approach brings speed of calculation and accuracy of the model. On the other hand, the presence of the [MASK] token during the training creates discrepancies with the fine-tuning phase in which this token is absent and it does not allow to formalize the co-occurrences of the hidden word with one's neighborhood. Despite these limitations, BERT is still the state-of-the-art pretraining approach based on AE (Yang et al. 2019).

BERT also exploits a second learning strategy called Next Sentence Prediction (NSP), and during each learning step, it relies on an average of both training strategies losses to optimize model parameters. This modality consists of predicting the sentence that logically follows the first one provided as input. In this way, it is possible to let the model also learn possible relations between sentences, for example, the textual

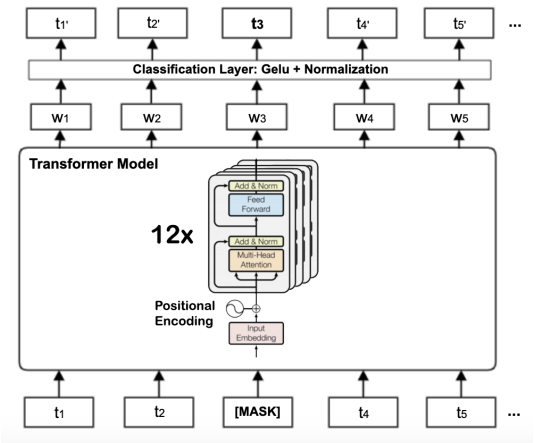


Figure 2
BERT and AlBERTo learning strategy

entailment. The two sentences are provided to the model separated by the token [SEP] and logically contextualized by adding to each embedding token the value of the specific sentence embedding, as well as the classic positional embedding. Then, the first sentence is analyzed and then a level of confidence determined to predict whether a given second hypothesized sentence in the pair "fits" logically as the proper next sentence, or not, with either a positive, negative, or neutral prediction, from a text collection under scrutiny.

The code in Listing 1 reports the instructions needed to launch the training phase of the BERT model on Colab, via TPU. Specifically, it starts by defining the model construction function through the *model_fn_builder* function. Among the most important hyper-parameters to be considered we have the learning rate that determines how fast the model changes its parameter weights, the number of training steps that specifies how many times the model must repeat the parameter optimization phase and the use_TPU flag to be sure that the model uses the TPU. After defining the model creation function, it is required to define the "RunConfig" configuration used during the initialization of the model. In it, we specify the location where the model will save the temporary data, after how many steps it will backup the model to disk and the TPU configuration to use. The following step is about the definition of the model estimator, which is the element that will perform the training operations and then optimize the model parameter weights. It uses the model creation feature and the "RunConfig" to perform the training of the model on a batch of examples at a time. In order to launch the training function from the estimator, it is necessary to define the input data loader function. The function requires the maximum input size and the maximum number of predictions to be performed for each sequence. In this phase, we have not yet investigated how to create the input data set, but this will be explained later in Section 3.2. With the functions defined in this section, we now have all the elements required to launch the training of BERT from scratch on a specific input data using the TPU.

The hyper-parameters used for configuring the model architecture of AlBERTo are reported in Listing 2.

Listing 1
Training code.

```
model_fn = model_fn_builder(
      bert_config=bert_config,
      init_checkpoint=INIT_CHECKPOINT,
      learning_rate=LEARNING_RATE,
      num_train_steps=TRAIN_STEPS,
      num_warmup_steps=10,
      use_tpu=USE_TPU,
      use_one_hot_embeddings=True)

tpu_cluster_resolver = tf.contrib.cluster_resolver.TPUClusterResolver(TPU_ADDRESS)

run_config = tf.contrib.tpu.RunConfig(
    cluster=tpu_cluster_resolver,
    model_dir=BERT_GCS_DIR,
    save_checkpoints_steps=SAVE_CHECKPOINTS_STEPS,
    tpu_config=tf.contrib.tpu.TPUConfig(
        iterations_per_loop=SAVE_CHECKPOINTS_STEPS,
        num_shards=NUM_TPU_CORES,
        per_host_input_for_training=tf.contrib.tpu.InputPipelineConfig.PER_HOST_V2))

estimator = tf.contrib.tpu.TPUEstimator(
    use_tpu=USE_TPU,
    model_fn=model_fn,
    config=run_config,
    train_batch_size=TRAIN_BATCH_SIZE,
    eval_batch_size=EVAL_BATCH_SIZE)

train_input_fn = input_fn_builder(
        input_files=input_files,
        max_seq_length=MAX_SEQ_LENGTH,
        max_predictions_per_seq=MAX_PREDICTIONS,
        is_training=True)

#RUN THE TRAINING
estimator.train(input_fn=train_input_fn, max_steps=TRAIN_STEPS)
```

Listing 2
BERT model configuration values.

```

bert_base_config = {
  "attention_probs_dropout_prob": 0.1,
  "directionality": "bidi",
  "hidden_act": "gelu",
  "hidden_dropout_prob": 0.1,
  "hidden_size": 768,
  "initializer_range": 0.02,
  "intermediate_size": 3072,
  "max_position_embeddings": 512,
  "num_attention_heads": 12,
  "num_hidden_layers": 12,
  "type_vocab_size": 2,
  "vocab_size": 128000
}
```

Listing 3
Training phase configuration values.

```
# Input data pipeline config
TRAIN_BATCH_SIZE = 128
MAX_PREDICTIONS = 20
MAX_SEQ_LENGTH = 128
MASKED_LM_PROB = 0.15

# Training procedure config
EVAL_BATCH_SIZE = 64
LEARNING_RATE = 2e-5
TRAIN_STEPS = 1000000
SAVE_CHECKPOINTS_STEPS = 2500
NUM_TPU_CORES = 8
```

As reported into the BERT official source repository we describe the parameters in Listing 2 as follow:

- **attention_probs_dropout_prob**: The dropout ratio for the attention probabilities.
- **hidden_act**: The non-linear activation function (function or string) in the encoder and pooler.
- **hidden_dropout_prob**: The dropout probability for all fully connected layers in the embeddings, encoder, and pooler.
- **hidden_size**: Size of the encoder layers and the pooler layer.
- **initializer_range**: The stdev of the truncated_normal_initializer for initializing all weight matrices.
- **intermediate_size**: The size of the "intermediate" (i.e., feed-forward) layer in the Transformer encoder.
- **max_position_embeddings**: The maximum sequence length that this model might ever be used with. Typically set this to something large just in case (e.g., 512 or 1024 or 2048).
- **num_attention_heads**: Number of attention heads for each attention layer in the Transformer encoder.
- **num_hidden_layers**: Number of hidden layers in the Transformer encoder.
- **type_vocab_size**: The vocabulary size of the ‘token_type_ids‘ passed into ‘BertModel‘.
- **vocab_size**: Vocabulary size of ‘inputs_ids‘ in ‘BertModel‘.

The parameters in Listing 3 are self-expressive, represent the batch size for training, the number of max predictions for each training example, the maximum size of the input, and the percentage of token masked during the model training. The training function has been launched on the Google Collaborative Environment (Colab) configured as previously described. In total, it took ~ 50 hours to create a complete AlBERTo model. More technical details are available in the Notebook *"Italian Pre-training BERT*

from scratch with cloud TPU" into the AlBERTo project repository on GitHub[5]. The final loss value obtained on training data is equal to 0.245. We do not format our data in order to have a sequence of tweets, and consequently, we do not perform the next sentence to predict training process such as well known in other language models such as RoBERTa (Liu et al. 2019).

3.2 Input Data processing

Once the learning strategy is defined, the consequent step is the preparation of the textual data to be used in the model. BERT's English model has been trained on text data containing no particular characters such as hashtags and mentions, so the pre-processing phase is implemented as a simple cleaning of the data from unexpected, accented, or incorrectly coded characters. In our case, the pre-processing phase is more complex, and further steps are indispensable.

More specifically, using Python as the programming language, two libraries were mainly adopted: Ekphrasis (Baziotis, Pelekis, and Doulkeridis 2017) and SentencePiece[6] (Kudo 2018). Ekphrasis is a popular tool comprising an NLP pipeline for text extracted from Twitter. It has been used for:

- Normalizing URL, emails, mentions, percents, money, time, date, phone numbers, numbers, emoticons;
- Tagging and unpacking hashtags.

The normalization phase consists in replacing each term with a fixed tuple $< [entity \quad type] >$. The tagging phase consists of enclosing hashtags with two tags $< hashtag > ... < /hashtag >$ representing their beginning and end in the sentence. The hashtags have also been unpacked. That means the entire world has been split, when possible, to the corresponding meaningful words. As an example, the hashtag #*bellaitalia* has been tagged and unpacked as "$< hashtag >$ bella italia $< /hashtag >$". This process was carried out in order to be able to treat hashtags as significant elements of the sentence, without forgetting their original role as a non-standard element of the sentence. The text is cleaned and made easily readable by the network by converting it to its lowercase form and all characters except emojis, !, ? and accented characters have been deleted. An example of pre-processed tweet is shown in Figure 3.

Original tweet: *#labuonascuola Eccolo, il rapporto on line qui http://t.co/U5AXNySoJu*

Preprocessed: <hashtag> la buona scuola </hashtag> eccolo il rapporto on line qui <url>

Figure 3
Example of preprocessed Tweet

The standardized text needs a tokenization approach so that it can be used correctly during the training. In particular, BERT uses the WordPiece tokenizer, not available as opensource. An efficient alternative is found in the use of SentencePiece[7],

5 https://github.com/marcopoli/AlBERTo-it
6 https://github.com/google/sentencepiece
7 https://github.com/google/sentencepiece

an unsupervised algorithm that uses a vocabulary of words for subdividing the text into tokens or subword units. It can process up to 50k sentences per second independently from the language of the text. The subdivision is based on a simple regularization method, namely subword regularization, which trains the model with multiple subword segmentations probabilistically sampled during the training (Kudo 2018). The construction of the vocabulary is performed on a portion equal to the 5% of the training dataset as a consequence of the high consumption of RAM of this process. The vocabulary generated for AlBERTo consists of 128.000 lower-case words, four times the size of BERT vocabulary. It includes the most common terms in the training set and the subwords which occur in the middle of words, annotating them with '##' in order to be able to encode also slang, incomplete, or uncommon words. An example of a piece of the vocabulary generated for AlBERTo is shown in Figure 4.

```
[PAD]  [UNK]  [CLS]  [SEP]  [MASK]
##>  <  ##hashtag  ##user  </  ##url
!  di  e  a  che  il  la  ##number
non  ?  è  per  anche  in  un  della
  l  ma  mi  i  grazie  tutti  alla
con  si  sono  una  tutto  le  ho
se ##👏 ##💘 ##😏 ##🙏 ##💋 ##😔
fare io  da ti bene fatto italia
```

Figure 4
An extract of the vocabulary created by SentencePiece for AlBERTo

The dataset used for the learning phase of AlBERTo is TWITA (Basile, Lai, and Sanguinetti 2018), a huge corpus of Tweets in the Italian language collected from February 2012 to the present day from Twitter's official streaming API. In our configuration, we randomly selected 200 million Tweets from 2013 to 2015, removing re-tweets, and we processed them with the pre-processing pipeline described previously. In total, we obtained 191GB of raw data. The standard format for providing text data as input to a BERT model is the TFRecord. This data format allows the input to be divided into tensorflow optimized records of the size of the shard. In AlBERTo the shard size is equal to 256000 tweets. For datasets that are too large to be stored fully in memory this is an advantage as only the data that is required at the time (e.g. a batch) is loaded from disk and then processed. A TFRecord file stores your data as a sequence of binary strings. This means you need to specify the structure of your data before you write it to the file. In particular the structure of the data, the percentage of masking for learning and the length of the input sentences is passed as parameter of the BERT "create_pretraining_data.py" class. The whole dataset transformation into TFRecords requires around 10 hours.

Listing 4
Creation of input tfrecords.

```
PRC_DATA_FPATH = "twita/download/twita_200M.txt"
!mkdir ./shards
!split -a 4 -l 256000 -d $PRC_DATA_FPATH ./shards/shard_
!ls ./shards/

MAX_SEQ_LENGTH = 128
MASKED_LM_PROB = 0.15
MAX_PREDICTIONS = 20
DO_LOWER_CASE = True
PROCESSES = 2
```

```
PRETRAINING_DIR = "pretraining_data"

XARGS_CMD = ("ls ./shards/ | "
             "xargs -n 1 -P {} -I{} "
             "python3 bert/create_pretraining_data.py "
             "--input_file=./shards/{} "
             "--output_file={}/{}.tfrecord "
             "--vocab_file={} "
             "--do_lower_case={} "
             "--max_predictions_per_seq={} "
             "--max_seq_length={} "
             "--masked_lm_prob={} "
             "--random_seed=34 "
             "--dupe_factor=5")

XARGS_CMD = XARGS_CMD.format(PROCESSES, '{}', '{}', PRETRAINING_DIR, '{}', VOC_FNAME,
    DO_LOWER_CASE, MAX_PREDICTIONS, MAX_SEQ_LENGTH, MASKED_LM_PROB)

tf.gfile.MkDir(PRETRAINING_DIR)
!$XARGS_CMD
```

As can be observed from the code reported in the Listing 4, we have kept the input text size at the standard value equal to 128 as a result of the shortness that each tweet achieves. We also left the masking percentage of the sentences fixed at the standard value of 15%. Finally, the model is trained without taking into account uppercase letters, thus resulting case-insensitive with consequent loss of representation power. The TFRecords produced in this way have therefore been used by the training routines already described previously.

3.3 Fine-tuning and Model release

The pre-trained model was released to the community through the GitHub platform. Specifically, the entire python code necessary to create a BERT model from scratch on your data and the code to use to perform the fine-tuning phase of the model in a specific application domain has been released. The pre-trained model is too general to be used directly in a classification task, so it needs to be refined to adapt its internal parameters to the domain and specific task. Fine-tuning involves copying the weights from a pre-trained network and tuning them on the downstream task.

Listing 5
AlBERTo Fine-Tuning for classification task.

```
f = lambda x: InputExample(guid=None,text_a = x[1],text_b = None,label = int(x[0]))
fine_tuning_examples = map(f,examples)

fine_tuning_features = convert_examples_to_features(
      fine_tuning_examples, label_list, MAX_SEQ_LENGTH, tokenizer)

train_input_fn = input_fn_builder(
    features=fine_tuning_features,
    seq_length=MAX_SEQ_LENGTH,
    is_training=True,
    drop_remainder=True)

estimator.train(input_fn=train_input_fn, max_steps=num_train_steps)
```

The code reported in Listing 5 shows how to perform the fine-tuning phase of AlBERTo in case of a classification task. It is important to notice that in the function *lambda*, the portion of text_b remains empty because we work on a classification task which does not require two sentences such as entailment or QA. Once the examples have been transformed into a BERT compatible format, defined the fine-tuning function

with the corresponding hyper-parameters, it is possible to perform the real fine-tuning training for a number of steps depending on the application domain. Usually, this value is between 3 and 10. After this step, it is possible to use the model for predictions with a similar strategy.

To facilitate the use of the model, we additionally decided to distribute it through the Transformers library [8] (Wolf et al. 2019). The huggingface Transformer library provides methods for using state of the art models, such as BERT, GPT-2, RoBERTa, XLM, DistilBert, XLNet, CTRL, and more. In particular, it can be used for Natural Language Understanding (NLU) and Natural Language Generation (NLG) tasks in more than 100 languages. The library is written in Python and is interoperable between TensorFlow 2.0 and PyTorch. The details about AlBERTo loaded in Transformers are available on-line[9].

Listing 6
Use of AlBERTo using Transformers library.

```
from tokenizer import *
from transformers import AutoTokenizer, AutoModel

a = AlBERTo_Preprocessing(do_lower_case=True)
s: str = "#IlGOverno presenta le linee guida sulla scuola #labuonascuola – http://t.co/SYS1T9QmQN"
b = a.preprocess(s)

tok = AutoTokenizer.from_pretrained("m-polignano-uniba/bert_uncased_L-12_H-768_A-12_italian_alb3rt0")
tokens = tok.tokenize(b)
print(tokens)

model = AutoModel.from_pretrained("m-polignano-uniba/bert_uncased_L-12_H-768_A-12_italian_alb3rt0")
```

The code reported in Listing 6 shows how it is possible to download and use AlBERTo by using the Transformers library with a few instructions. It is important to underline that as first instruction we load a package called "tokenizer" this has been explicitly created for AlBERTo and distributed through the GitHub repository in order to perform pre-processing operations on the text that are compliant with the AlBERTo model, including the transformation of hashtags, mentions, URL, etc.. After this pre-processing you can load the model with a simple instruction.

Finally, we decided to make available our fine-tuned AlBERTo models through RESTful APIs for free, to use the classification models already implemented through BERT. In particular, they concern the tasks described in Section 4 (subjectivity, polarity, irony, hate speech). Since we cannot guarantee their efficiency and stability on a very high number of calls, we do not publicly release the endpoint IP address, but provide access keys by request.

4. Evaluation of AlBERTo on NLP tasks

We evaluate AlBERTo on two publicly available benchmarks on the Italian language. The first is the dataset released for the SENTIPOLC (SENTIment Polarity Classification)

8 A special thanks to Angelo Basile (angelo.basile@symanto.net), Junior Research Scientist at Symanto, for providing the models transformed into PyTorch and directly shareable through Transformers released by huggingface.co: https://huggingface.co/.

9 https://huggingface.co/m-polignano-uniba/bert_uncased_L-12_H-768_A-12_italian_alb3rt0

Table 1
Results obtained using the official evaluation script of SENTIPOLC 2016

	Prec. 0	**Rec. 0**	**F1. 0**
Subjectivity	0.6838	0.8058	0.7398
Polarity Pos.	0.9262	0.8301	0.8755
Polarity Neg.	0.7537	0.9179	0.8277
Irony	0.9001	0.9853	0.9408
	Prec. 1	**Rec. 1**	**F1 . 1**
Subjectivity	0.8857	0.8015	0.8415
Polarity Pos.	0.5818	0.5314	0.5554
Polarity Neg.	0.7988	0.5208	0.6305
Irony	0.6176	0.1787	0.2772

shared task (Barbieri et al. 2016) carried out at EVALITA 2016 (Basile et al. 2016), a challenge on sentiment analysis on Italian tweets. The second is the hate speech-annotated corpus released for the HaSpeeDe shared task (Bosco et al. 2018) held at EVALITA 2018 (Caselli et al. 2018), a challenge on the detection of hateful content in Italian social media. We verified that the texts contained in those datasets come from a distribution different from the ones used for the pre-training of AlBERTo.

4.1 Sentiment Analysis

The SENTIPOLC challenge includes three subtasks:

- **Subjectivity Classification**: “a system must decide whether a given message is subjective or objective";
- **Polarity Classification**: “a system must decide whether a given message is of positive, negative, neutral or mixed sentiment";
- **Irony Detection**: “a system must decide whether a given message is ironic or not".

Data provided for training and test are tagged with six fields containing values related to manual annotation: subj, opos, oneg, iro, lpos, lneg. These labels indicate if the sentence is subjective, positive, negative, ironical, literal positive, and literal negative, respectively. For each of these classes, there is a 1 where the sentence satisfy the label, a 0 otherwise.
The last two labels “lpos" and “lneg" that describe the literal polarity of the tweet have not been considered in the current evaluation (nor in the official shared task evaluation). In total, 7,410 tweets have been released for training and 2,000 for testing. We do not used any validation set because we do not performed any phase of model selection during the fine-tuning of AlBERTo. The evaluation was performed considering precision (p), recall (r) and F1-score (F1) for each class and for each classification task.

AlBERTo fine-tuning. We fine-tuned AlBERTo four different times, in order to obtain one classifier for each task except for the polarity where we have two of them. In particular,

Table 2
Comparison of results with the best systems of SENTIPOLC for subjectivity classification task

System	Obj	Subj	F
AlBERTo	***0.7398***	***0.8415***	***0.7906***
Unitor.1.u	0.6784	0.8105	0.7444
Unitor.2.u	0.6723	0.7979	0.7351
samskara.1.c	0.6555	0.7814	0.7184
ItaliaNLP.2.c	0.6733	0.7535	0.7134
BERT Multilang	*0.4765*	*0.5197*	*0.4981*

Table 3
Comparison of results with the best systems of SENTIPOLC for polarity classification task

System	Pos	Neg	F
AlBERTo	***0.7155***	***0.7291***	***0.7223***
UniPI.2.c	0.6850	0.6426	0.6638
Unitor.1.u	0.6354	0.6885	0.6620
Unitor.2.u	0.6312	0.6838	0.6575
ItaliaNLP.1.c	0.6265	0.6743	0.6504
BERT Multilang	*0.5511*	*0.4978*	*0.5230*

Table 4
Comparison of results with the best systems of SENTIPOLC for irony classification task

System	Non-Iro	Iro	F
AlBERTo	***0.9408***	***0.2772***	***0.6090***
tweet2check16.c	0.9115	0.1710	0.5412
CoMoDI.c	0.8993	0.1509	0.5251
tweet2check14.c	0.9166	0.1159	0.5162
IRADABE.2.c	0.9241	0.1026	0.5133
BERT Multilang	*0.9376*	*0.0000*	*0.4688*

we created one classifier for the Subjectivity Classification, one for Polarity Positive, one for Polarity Negative and one for the Irony Detection. Each time we have re-trained the model for three epochs, using a learning rate of 2e-5 with 1000 steps per loops on batches of 512 example from the training set of the specific task. For the fine-tuning of the Irony Detection classifier, we increased the number of epochs of training to ten observing low performances using only three epochs as for the other classification tasks. The fine-tuning process lasted $\sim$ 4 minutes every time.

Discussion of results. The results reported in Table 1 show the output obtained from the official evaluation script of SENTIPOLC 2016. It is important to note that the values on the individual classes of precision, recall and, F1 are not compared with those of the systems that participated in the competition because they are not reported in the overview paper of the task. Nevertheless, some considerations can be drawn. The classifier based on AlBERTo achieves, on average, high recall on class 0 and low values on class 1. The opposite situation is instead observed on the precision, where for the

class 1 it is on average superior to the recall values. This suggests that the system is very good at classifying a phenomenon and when it does, it is sure of the prediction made even at the cost of generating false negatives. Interesting is to compare AlBERTo results with them of *BERT Multilang*. In particular, this configuration is using a standard BERT small model uncased pre-trained for multilingual purposes (102 languages, 12-layer, 768-hidden, 12-heads, 110M parameters). The model obtains results lower than all the other participants at the tasks, including AlBERTo. These results are motivated by the strategy used for pre-training that model, i.e., using mixed-language corpora. This decision makes the model applicable universally on a large number of languages and NLP tasks but, at the same time, less performant than one focused on only a single language.

On each of the sub-tasks of SENTIPOLC (Table 2-4), it can be observed that AlBERTo has obtained state of the art results without any heuristic tuning of learning parameters (model as it is after fine-tuning training) except in the case of irony detection where it was necessary to increase the number of epochs of the learning phase of fine-tuning. Comparing AlBERTo with the best system of each subtask, we observe an increase in results between 7% and 11%. The results obtained are exciting, from our point of view, for further future work.

4.2 Hate Speech Detection

The HaSpeeDe evaluation campaign was proposed to create a benchmark for Hate Speech (HS) detection in the Italian language. The shared task was carried out by dividing the problem into four different tasks:

- **HaSpeeDe-FB**: where the goal is to train the model and predict if the contents are HS on data extracted from Facebook;
- **HaSpeeDe-TW**: where the goal is to train the model and predict if the contents are of HS on data extracted from Twitter;
- **Cross-HaSpeeDe_FB**: where the goal is to train the model on data collected from Facebook and predict if the contents are of HS on data extracted from Twitter;
- **Cross-HaSpeeDe_TW**: where the goal is to train the model on data collected from Twitter and predict if the contents are of HS on data extracted from Facebook;

It is interesting to note that in the first two tasks, the model must be able to classify data coming from the same information source as the training phase. Unlike the two "Cross" tasks, the data to be classified are different from those used for the test, making the task of the classifier more challenging due to the differences in writing styles of the two platforms. In fact, not only are Twitter data shorter, containing mentions, hashtags, and retweets, but overall, they contain less HS than Facebook data (only 32% compared to 68% for Facebook).

The **Facebook dataset** is collected from public pages on Facebook about newspapers, public figures, artists and groups on heterogeneous topics. More than 17,000 comments were collected from 99 posts and subsequently annotated by 5 bachelor students. The final dataset released consists of 3,000 training phrases (1,618 not HS, 1,382 HS) and

1000 test phrases (323 not HS, 677 HS).

The **Twitter dataset** is part of the Hate Speech Monitoring program, coordinated by the Computer Science Department of the University of Turin with the aim of detecting, analyzing and countering HS with an inter-disciplinary approach (Bosco et al. 2017). Data were collected using keywords related to the concepts of immigrants, Muslims and Roma. Data are annotated partly by experts and partly by crowdsourcing. Also for this dataset 3,000 training tweets (2,028 not HS and 972 HS) and 1,000 test tweets (676 not HS and 324 HS) were released.

The evaluation metrics used in HaSpeeDe are precision, recall and F1-measure. Since the two classes (HS and not HS) are unbalanced within the datasets, the F1 metric has been calculated separately on the two classes and then macro-averaged. For all tasks, the baseline score has been computed as the performance of a classifier based on the most frequent class.

HaSpeeDe has received strong participation from the scientific community and therefore a large number of solutions to the task have been proposed, including Support Vector Machine, deep learning (mostly Bi-LSTM networks and convolutional neural networks), and ensemble models.

AlBERTo-HS fine-tuning. We fine-tuned AlBERTo two different times, in order to obtain one classifier for each different dataset available as a training set. In particular, we created one classifier for the HaSpeeDe-FB and the Cross-HaSpeeDe_FB tasks using Facebook training data and one for the HaSpeeDe-TW and the Cross-HaSpeeDe_TW using the Twitter training set. The fine-tuning learning phase has been run for 15 epochs, using a learning rate of 2e-5 with 1,000 steps per loops on batches of 512 examples. The fine-tuning process lasted $\sim$ 4 minutes every time.

Discussion of results. The evaluation of the results obtained by the AlBERTo-HS classifier was carried out using the official evaluation script released at the end of the campaign [10]. Consequently, all the results obtained are replicable and comparable with those present in the final ranking of HaSpeeDe.

From the tables of results (Tables 5–8), it is possible to observe how AlBERTo-HS succeeds in obtaining a state of the art results for two tasks out of four. The differences with other systems proposed in the evaluation campaign are about its simplicity to be applied. A simple fine-tuning phase of AlBERTo on domain data allows us to obtain very encouraging results. It is also noticeable that the entire process of pre-processing and fine-tuning takes just a few minutes. In particular, the model is able to adapt in an excellent way to annotated data (although with the risk of overfitting) producing excellent results if used in the same application domain of the tuning phase. This is the case with the results obtained for the HaSpeeDe-FB and HaSpeeDe-TW tasks.

Looking at the results obtained for the classification of data coming from Facebook (Tab. 5), it is possible to observe how the classifier is able to capture the characteristics of the social language through the fine-tuning phase. In particular, it is able to move its learned weights from them obtained parsing the original training language based on Twitter to the one used on Facebook. AlBERTo-HS obtains better performances than those of other participants in the evaluation campaign, with respect to the precision in

10 http://www.di.unito.it/~tutreeb/haspeede-evalita18/data.html

Table 5
Results of the HaSpeeDe-FB task

	NOT HS			HS			
	Precision	Recall	F-score	Precision	Recall	F-score	***Macro-Avg F-score***
most_freq							0.2441
AlBERTo-HS	**0.8603**	0.7058	**0.7755**	0.8707	**0.9453**	**0.9065**	**0.8410**
ItaliaNLP 2	0.8111	0.7182	0.7619	0.8725	0.9202	0.8957	0.8288
InriaFBK 1	0.7628	0.6873	0.7231	0.8575	0.8980	0.8773	0.8002
Perugia 2	0.7245	0.6842	0.7038	0.8532	0.8759	0.8644	0.7841
RuG 1	0.699	0.6904	0.6947	0.8531	0.8581	0.8556	0.7751
HanSEL	0.6981	0.6873	0.6926	0.8519	0.8581	0.8550	0.7738
VulpeculaTeam	0.6279	0.7523	0.6845	0.8694	0.7872	0.8263	0.7554
RuG 2	0.6829	0.6068	0.6426	0.8218	0.8655	0.8431	0.7428
GRCP 2	0.6758	0.5294	0.5937	0.7965	0.8788	0.8356	0.7147
StopPropagHate 2	0.4923	0.6965	0.5769	0.8195	0.6573	0.7295	0.6532
Perugia 1	0.3209	0.9907	0.4848	0.0000	0.0000	0.0000	0.2424

Table 6
Results of the HaSpeeDe-TW task

	NOT HS			HS			
	Precision	Recall	F-score	Precision	Recall	F-score	**Macro-Avg F-score**
most_freq							0.4033
AlBERTo-HS	0.8746	0.8668	0.8707	0.7272	0.7407	**0.7339**	**0.8023**
ItaliaNLP 2	0.8772	0.8565	0.8667	0.7147	0.75	0.7319	0.7993
RuG 1	0.8577	0.8831	0.8702	0.7401	0.6944	0.7165	0.7934
InriaFBK 2	0.8421	0.8994	0.8698	0.7553	0.6481	0.6976	0.7837
sbMMMP	0.8609	0.852	0.8565	0.6978	0.7129	0.7053	0.7809
VulpeculaTeam	0.8461	0.8786	0.8621	0.7248	0.6666	0.6945	0.7783
Perugia 2	0.8452	0.8727	0.8588	0.7152	0.6666	0.6900	0.7744
StopPropagHate 2	0.8628	0.7721	0.8149	0.6101	0.7438	0.6703	0.7426
GRCP 1	0.7639	0.8713	0.8140	0.6200	0.4382	0.5135	0.6638
HanSEL	0.7541	0.8801	0.8122	0.6161	0.4012	0.4859	0.6491

identifying the not-HS posts (0.8603), and the recall of the HS posts (0.9453). The high recall for hate messages allows us to assume that, on Facebook, they are characterized by specific topics that make the classification task more inclusive at the cost of accuracy, especially when not explicit hate messages are faced. As an example, the message "Comunque caro Matteo se non si prendono provvedimenti siamo rovinati." (*However dear Matteo if we do not do something we are ruined*) is classified as a hate message even if the annotators have considered it to be not a hate message. In this example, it is arguable whether a component of hate is present in the intent of the writer, even if it is not overt in what they write. In other cases, words like "severe" (plural form of *severe, strict*) have tricked the model into classifying clearly neutral messages like the following as hate messages: "Matteo sei la nostra voce!!! Noi donne non possiamo fare un cavolo! !! Leggi più severe!" (*Matteo you are our voice!!! Us women cannot do anything!!! Stricter laws!*). Nevertheless, the average F1 score higher than 0.8410, show us that, unlike in Twitter, the use of more characters available for writing allows people to be more verbose and, therefore, more comfortable to identify. Table 6 shows the results obtained for the classification of tweets. Here the values are not so different from the top ranking system in the evaluation campaign, even if the average value of F1 obtained of 0.8023 proves

Table 7
Results of the Cross-HaSpeeDe_FB task

	NOT HS			HS			
	Precision	Recall	F-score	Precision	Recall	F-score	**Macro-Avg F-score**
most_freq							0.4033
InriaFBK 2	0.8183	0.6597	0.7305	0.4945	0.6944	0.5776	0.6541
VulpeculaTeam	0.8181	0.6390	0.7176	0.4830	0.7037	0.5728	0.6452
Perugia 2	0.8503	0.5547	0.6714	0.4615	0.7962	0.5843	0.6279
ItaliaNLP 1	0.9101	0.4644	0.6150	0.4473	0.9043	0.5985	0.6068
GRCP 2	0.7015	0.7928	0.7444	0.4067	0.2962	0.3428	0.5436
RuG 1	0.8318	0.4023	0.5423	0.3997	0.8302	0.5396	0.5409
AlBERTo-HS	0.8955	0.2662	0.4104	0.3792	0.9351	0.5396	0.4750
HanSEL	0.7835	0.2677	0.3991	0.3563	0.8456	0.5013	0.4502
StopPropagHate	0.6579	0.3727	0.4759	0.3128	0.5956	0.4102	0.4430

Table 8
Results of the Cross-HaSpeeDe_TW task

	NOT HS			HS			
	Precision	Recall	F1-score	Precision	Recall	F1-score	**Macro F1-score**
most_freq							0.2441
ItaliaNLP 2	0.5393	0.7647	0.6325	0.8597	0.6883	0.7645	0.6985
AlBERTo-HS	0.5307	0.7492	0.6213	0.8511	0.6838	0.7583	0.6898
InriaFBK 2	0.5368	0.6532	0.5893	0.8154	0.7311	0.771	0.6802
VulpeculaTeam	0.4530	0.7461	0.5637	0.8247	0.5701	0.6742	0.6189
RuG 1	0.4375	0.6934	0.5365	0.7971	0.5745	0.6678	0.6021
HanSEL	0.3674	0.8235	0.5081	0.7934	0.3234	0.4596	0.4838
Perugia 2	0.3716	0.9318	0.5313	0.8842	0.2481	0.3875	0.4594
GRCP 1	0.3551	0.8575	0.5022	0.7909	0.2570	0.3879	0.4451
StopPropagHate	0.3606	0.9133	0.5170	0.8461	0.2274	0.3585	0.4378

to be the best. This suggests that the presence in the tweets of particular characters and implicitly of hate, the brevity of the latter, and the increase in the number of ironic tweets make the task more complicated than the previous one.

As far as "Cross" classification problems are concerned, the results are not guaranteed. In Table 7 it can be observed that the model has not been able to correctly abstract from the domain data, obtaining not very good results for the classification in a different domain. In particular, the model trained on Facebook is able to obtain a score of 0.4750 of F1 on Twitter test data. A similar situation is repeated for the results in Table 8 where for the task Cross-HaSpeeDe_TW the model is able to generalize slightly better than before but still gets the second place in the ranking. These results confirm the difficulty of the Cross tasks and the drop in performance that is obtained through a transfer-learning strategy like the one adopted here. The great differences in writing styles used on the two social networks do not allow the model to adapt properly to the domain of application if fine-tuned on different stylistic data. So that AlBERTo is not able to grasp those particularities of the language to be used in the classification phase.

5. Conclusion

In this work, we described AlBERTo, the first Italian language model based on social media writing style. The model has been trained using the official BERT source code on a Google TPU-V2 relying on 200M tweets in the Italian language. The pre-trained model has been fine-tuned on the data available for the classification tasks SENTIPOLC 2016 (polarity and irony) and HaSpeeDe (hate speech detection), showing SOTA results in both benchmarks. We facilitate the reuse of AlBERTo by publishing the trained model and the source code on GitHub [11], on the HuggingFace repository [12], and via a HTTP REST webservice.

The results allow us to promote AlBERTo as the starting point for future research in this direction. Since our results showed has it is possible to obtain an excellent result in classification by merely carrying out a phase of fine-tuning the model, we will consider making a further comparison with other language understanding models such as GPT2, XLNet, RoBERTa trained on the Italian language with the aim of verifying if they can be more robust to the changes in the writing style of the text to be classified. We are also considering the possibility of developing a different version of AlBERTo trained on Wikipedia. Furthermore, we are working on the integration of AlBERTo into the national project, "Contro l'odio"[13], that aims to monitor, classify and summarize in statistics the hate messages in Italian identified via Twitter. In this direction, we started an experimental line of research to leverage AlBERTo in diachronic classification tasks, where the language model trained on a large-scale dataset is showing promising results towards stabilizing the prediction capability over time.

Acknowledgment

The work of Marco de Gemmis is funded by project POR Puglia FESR FSE 2014-2020 - Sub-Azione 1.4.B progetto Feel at Home codice raggruppamento: $UIKTJF3$. Dominio di riferimento: Smart Cities & Communities. The work of Valerio Basile is partially funded by Progetto di AteneoCSP 2016 (*Immigrants, Hate and Prejudice in Social Media*, S1618_L2_BOSC_01.

References

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, Francesco 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)*, volume 1749 of *CEUR Workshop Proceedings*, Naples, Italy, December 5-7. CEUR-WS.org.

Basile, Pierpaolo, Franco Cutugno, Malvina Nissim, Viviana Patti, Rachele Sprugnoli, et al. 2016. Evalita 2016: Overview of the 5th evaluation campaign of natural language processing and speech tools for italian. In *3rd Italian Conference on Computational Linguistics, CLiC-it 2016 and 5th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, EVALITA 2016*, volume 1749, pages 1–4, Naples, Italy, December, 7. CEUR-WS.

Basile, Valerio, Mirko Lai, and Manuela Sanguinetti. 2018. Long-term social media data collection at the university of turin. In Elena Cabrio, Alessandro Mazzei, and Fabio Tamburini,

11 https://github.com/marcopoli/AlBERTo-it

12 https://huggingface.co/m-polignano-uniba/bert_uncased_L-12_H-768_A-12_italian_alb3rt0

13 https://controlodio.it/

editors, *Proceedings of the Fifth Italian Conference on Computational Linguistics (CLiC-it 2018)*, volume 2253 of *CEUR Workshop Proceedings*, Torino, Italy, December 10-12. CEUR-WS.org.

Baziotis, Christos, Nikos Pelekis, and Christos Doulkeridis. 2017. Datastories at semeval-2017 task 4: Deep lstm with attention for message-level and topic-based sentiment analysis. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 747–754, Vancouver, Canada, August. Association for Computational Linguistics.

Bojanowski, Piotr, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Bosco, Cristina, Felice Dell'Orletta, Fabio Poletto, Manuela Sanguinetti, and Maurizio Tesconi. 2018. Overview of the EVALITA 2018 hate speech detection task. In Tommaso Caselli, Nicole Novielli, Viviana Patti, and Paolo Rosso, editors, *Proceedings of the Sixth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2018) co-located with the Fifth Italian Conference on Computational Linguistics (CLiC-it 2018)*, volume 2263 of *CEUR Workshop Proceedings*, Turin, Italy, December 12-13. CEUR-WS.org.

Bosco, Cristina, Viviana Patti, Marcello Bogetti, Michelangelo Conoscenti, Giancarlo Francesco Ruffo, Rossano Schifanella, and Marco Stranisci. 2017. Tools and resources for detecting hate and prejudice against immigrants in social media. In *Proceedings of AISB Annual Convention 2017*, pages 79–84, Bath, United Kingdom, April. AISB.

Buciluǎ, Cristian, Rich Caruana, and Alexandru Niculescu-Mizil. 2006. Model compression. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, page 535–541, New York, NY, USA. Association for Computing Machinery.

Caselli, Tommaso, Nicole Novielli, Viviana Patti, and Paolo Rosso. 2018. Evalita 2018: Overview on the 6th evaluation campaign of natural language processing and speech tools for italian. In Tommaso Caselli, Nicole Novielli, Viviana Patti, and Paolo Rosso, editors, *Proceedings of the Sixth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2018) co-located with the Fifth Italian Conference on Computational Linguistics (CLiC-it 2018)*, volume 2263 of *CEUR Workshop Proceedings*, Turin, Italy, December 12-13. CEUR-WS.org.

Chelba, Ciprian, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. 2014. One billion word benchmark for measuring progress in statistical language modeling. In Haizhou Li, Helen M. Meng, Bin Ma, Engsiong Chng, and Lei Xie, editors, *INTERSPEECH 2014, 15th Annual Conference of the International Speech Communication Association*, pages 2635–2639, Singapore, September 14-18. ISCA.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Kudo, Taku. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Volume 1: Long Papers*, pages 66–75, Melbourne, Australia, July 15-20. Association for Computational Linguistics.

Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems*, pages 3111–3119, Lake Tahoe, Nevada, United States, December 5-8.

Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543, Doha, Qatar, October 25-29. ACL.

Peters, Matthew E., Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In Marilyn A. Walker,

Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, USA, June 1-6. Association for Computational Linguistics.

Polignano, Marco, Pierpaolo Basile, Marco de Gemmis, and Giovanni Semeraro. 2019a. A comparison of word-embeddings in emotion detection from text using bilstm, CNN and self-attention. In George Angelos Papadopoulos, George Samaras, Stephan Weibelzahl, Dietmar Jannach, and Olga C. Santos, editors, *Adjunct Publication of the 27th Conference on User Modeling, Adaptation and Personalization, UMAP 2019*, pages 63–68, Larnaca, Cyprus, June 09-12. ACM.

Polignano, Marco, Pierpaolo Basile, Marco de Gemmis, and Giovanni Semeraro. 2019b. Hate speech detection through alberto italian language understanding model. In Mehwish Alam, Valerio Basile, Felice Dell'Orletta, Malvina Nissim, and Nicole Novielli, editors, *Proceedings of the 3rd Workshop on Natural Language for Artificial Intelligence co-located with the 18th International Conference of the Italian Association for Artificial Intelligence (AIIA 2019)*, volume 2521 of *CEUR Workshop Proceedings*, Rende, Italy, November 19th-22nd. CEUR-WS.org.

Polignano, Marco, Pierpaolo Basile, Marco de Gemmis, Giovanni Semeraro, and Valerio Basile. 2019c. Alberto: Italian BERT language understanding model for NLP challenging tasks based on tweets. In Raffaella Bernardi, Roberto Navigli, and Giovanni Semeraro, editors, *Proceedings of the Sixth Italian Conference on Computational Linguistics*, volume 2481 of *CEUR Workshop Proceedings*, Bari, Italy, November 13-15. CEUR-WS.org.

Sanh, Victor, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Solaiman, Irene, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, and Jasmine Wang. 2019. Release strategies and the social impacts of language models. *CoRR*, abs/1908.09203.

Sun, Yu, Shuohuan Wang, Yu-Kun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2020. ERNIE 2.0: A continual pre-training framework for language understanding. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020*, pages 8968–8975, New York, NY, USA, February 7-12. AAAI Press.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems*, pages 5998–6008, Long Beach, CA, USA, 4-9 December.

Wang, Alex, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In Tal Linzen, Grzegorz Chrupala, and Afra Alishahi, editors, *Proceedings of the Workshop: Analyzing and Interpreting Neural Networks for NLP, BlackboxNLP@EMNLP 2018*, pages 353–355, Brussels, Belgium, November 1. Association for Computational Linguistics.

Wolf, Thomas, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

Yang, Zhilin, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019*, pages 5754–5764, Vancouver, BC, Canada, 8-14 December.

Zhu, Yukun, Ryan Kiros, Richard S. Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *2015 IEEE International Conference on Computer Vision, ICCV 2015*, pages 19–27, Santiago, Chile, December 7-13. IEEE Computer Society.

Using Deep Neural Networks for Smoothing Pitch Profiles in Connected Speech

Michele Ferro*
CELI Language Technology

Fabio Tamburini**
Università di Bologna

This paper presents a new pitch tracking smoother based on deep neural networks (DNN). It leverages Long Short-Term Memories, a particular kind of recurrent neural network, for correcting pitch detection errors produced by state-of-the-art Pitch Detection Algorithms. The proposed system has been extensively tested using two reference benchmarks for English and exhibited very good performances in correcting pitch detection algorithms outputs when compared with the gold standard obtained with laryngographs.

1. Introduction

The pitch, and in particular the fundamental frequency - F0 - which represents its physical counterpart, is one of the most relevant perceptual parameters of the spoken language and one of the fundamental phenomena to be carefully considered when analysing linguistic data at a phonetic and phonological level. As a consequence, the automatic extraction of F0 has been a subject of study for a long time inspiring many works that aim to develop algorithms, commonly known as Pitch Detection Algorithms (PDA), able to reliably extract F0 from the acoustic component of the utterances.

Technically, the extraction of F0 is a problem far from trivial and the great variety of methodologies applied to this task demonstrate its extreme complexity, especially considering that it is difficult to design a PDA that works optimally for the different recording conditions, considering that parameters such as speech type, noise, overlaps, etc. are able to heavily influence the performances of this kind of algorithms.

Scholars worked hard searching for increasingly sophisticated techniques for these specific cases, although extremely relevant for the construction of real applications, considering solved, or perhaps simply abandoning, the problem of the F0 extraction for the so-called "clean speech". However, anyone who has used the most common programs available for the automatic extraction of F0 is well aware that errors of halving or doubling of the value of F0, to cite only one type of problem, are rather common and that the automatic identification of voiced areas within the utterance still poses numerous problems.

Every work that proposes a new method for the automatic extraction of F0 should accomplish an evaluation of the performances obtained in relation to other PDAs, but, usually, these assessments suffer from the typical shortcomings deriving from evaluation systems: they usually examine a very limited set of algorithms, often not available in their implementation, typically considering corpora not distributed, related to specific languages and/or that contain particular typologies of spoken language (pathological, disturbed by noise, overlapped dialogues, singing voices, etc.) (Veprek and Scordilis 2002; Wu, Wang, and Brown 2003; Kotnik, Höge, and Kacic

* CELI, Via S. Quintino, 31, 10121 Torino, Italy. E-mail: lele.ferro4@gmail.com
** Dept. of Classic Philology and Italian Studies, Via Zamboni 32, 40126 Bologna, Italy. E-mail: fabio.tamburini@unibo.it

2006; Jang et al. 2007; Luengo et al. 2007; Chu and Alwan 2009; Bartosek 2010; Huang and Lee 2012; Chu and Alwan 2012; Babacan et al. 2013; Gawlik and Wszolek 2018). There are a few studies, among the most recent, that have performed quite complete evaluations that are based on standard speech corpora often freely downloadable (de Cheveigné and Kawahara 2002; Camacho 2007; Wang and Loizou 2012; Sukhostat and Imamverdiyev 2015; Jouvet and Laprie 2017). Most research works use a single metric in the assessment that measures a single type of error, not considering or partly considering the whole panorama of indicators developed from the pioneering work of Rabiner and colleagues (1976) and therefore, in our opinion, the results obtained seem to be rather partial.

Tamburini (2013) performed an in-depth study of the different performances exhibited by several widely used PDAs by using standard evaluation metrics and well-established corpus benchmarks.

Starting from this study, the main purpose of our research was to improve the performances of the best Pitch Detection Algorithms identified in (Tamburini 2013) by introducing a post-processing smoother. In particular, we implemented a pitch smoother adopting Keras[1], a powerful high-level neural networks Application Program Interface (API), written in Python and able to run on top of TensorFlow, one of the most powerful machine learning libraries especially devoted to the development of large neural network models.

The paper is organised as follows: in Section 2 we will describe the pitch smoothing problem; in Section 3 we will present our neural PDA smoother while in section 4 we will define the experiments we did to evaluate our proposal; Section 5 shows the results and in Section 6 we will draw some provisional conclusions and propose some future works.

2. Pitch error correction and smoothing

Typical PDAs are organised into two different modules: the first stage tries to detect pitch frequencies frame by frame and, in the second stage, the pitch candidates, along with their probabilities, are connected into pitch contours using dynamic programming techniques (Bagshaw 1994; Chu and Alwan 2012; Gonzalez and Brookes 2014) or hidden Markov models (HMMs) (Jin and Wang 2011; Wu, Wang, and Brown 2003). In this second stage, the different PDAs apply various techniques in order to correct the intonation profile removing various errors produced by the first step.

These techniques are, however, not completely satisfactory and various types of errors remain in the intonation profile. That is why in the literature we can find several studies aiming at proposing pitch profile smoothers that further post-process the PDAs output trying to enhance the profile correctness. Some works try to correct intonation profiles by applying traditional techniques (Zhao, O'Shaughnessy, and Minh-Quang 2007; So, Jia, and Cai 2012; Jlassi, Bouzid, and Ellouze 2016), while few others (see for example (Kellman and Morgan 2017; Han and Wang 2014)) are based on DNN (either Multi-Layer Perceptrons or Elman Recurrent Neural Networks).

A complex periodic sound will actually have multiple repeating patterns in its waveform: some repeating at faster rates and some taking longer to repeat their cycles. It is the slowest (the longest period/lowest frequency) repeating pattern in a complex periodic sounds that governs the signal's perceived pitch. It is important mentioning the difference between perceptual and quantitative properties. Starting from this contrast, the pitch of a sound can be defined as the mental sensation or perceptual correlate of fundamental frequency; in general, if a sound has a higher fundamental frequency we perceive it as having a higher pitch. The relationship is not

1 https://keras.io/

linear, since human hearing has different responses for different frequencies. Roughly speaking, human pitch perception is most accurate between 100 Hz and 1000 Hz, and in this range pitch correlates linearly with frequency. Human hearing represents frequencies above 1000 Hz less accurately and above this range pitch correlates logarithmically with frequency.

F0 can be seen as the minimum frequency of the vocal folds vibration, or the frequency of the complex wave. All complex periodic sounds or waves can be mathematically analyzed as being composed of multiple single-frequency sounds/waves, such a series of sine waves: the Fourier's theorem states that any periodic signal is composed of the summation of multiple sine waves with particular amplitudes and phases. Fourier's theorem by extension implies that we can decompose complex periodic sounds into simple components (Byrd and H.Mintz 2010). The frequencies of a signal's harmonics are integer multiples of its F0: for this reason the second harmonic is 2 x F0, the third harmonic is 3 x F0 and so on. We cannot tell simply by looking at a complex waveform what its component frequencies or harmonics are. A computer is generally used to implement algorithms based on Fourier's theorem to find a complex signal's harmonics. A different kind of display, called a power spectrum, can be useful for showing the frequency composition or spectrum of a sound frame. A power spectrum, like in Figure 1, plots frequency on the horizontal axis and amplitude (or magnitude) on the vertical axis.

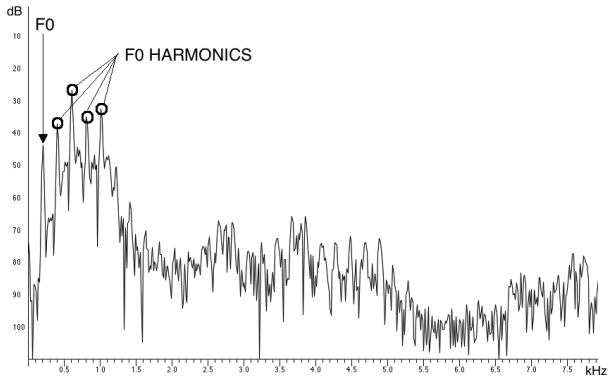


Figure 1
Power spectrum of a speech sample frame showing F0 and its harmonics.

Despite the number of studies devoted to the design of efficient PDAs, correct pitch extraction remains an open problem for various reasons. Pitch estimation, indeed, is a process heavily influenced by phenomena observed in spontaneous speech:

- F0 varies in time, potentially at each period of vibration of vocal folds;
- it often happens that "true" F0 has sub-harmonics as its submultiples which alter estimation of values in contrast with perception;
- the presence of resonances and filters in the vocal tract can emphasize harmonics of F0 multiples of the real value;
- sonority is often very irregular at the beginning and at the end of a voiced linguistic segment and all the frames involved in these transitions have minimal similarities between the corresponding waveforms;
- even for human experts the classification of the boundaries of voiced areas is a far from an easy task;

- due to certain disturbances it is possible that signals occur with a relevant percentage of periodicity in unvoiced areas too;
- voiced regions have a wide dynamic range of amplitude;
- it is difficult to distinguish periodic background noise from breathy voice;
- some voiced intervals are very short and they can be composed of just two or three cycles.

These different and complex problems have determined the spread of studies about F0 detection. We will focus on some of these algorithms later in this contribution.

The range of fundamental frequencies found in human voices is roughly 60 to 500 Hz, but in adult males a typical F0 might be 120 Hz; in a female voice a typical F0 might be 225 Hz, and in a child it might be 265 Hz. It is worth underlying that variation in fundamental frequency in speech is due to the structure of the larynx and the vocal folds only (Byrd and H.Mintz 2010). In addition to voicing, there are many ways to generate noise or sources of sound in the vocal tract during speech. For example, a fricative consonant creates noise by the turbulent airflow generated when air is forced through a narrow constriction, sometimes directed against the teeth as an obstacle. In this case, unlike the voicing source, the acoustic energy is generated in the mouth, not at the larynx. We state this because it has to be understood that many sound sources occur in speech, such as the noise created when a stop constriction is opened, but we will concentrate on the main sound source in speech - the voicing source - and look next at how the harmonic structure of this source is shaped by the vocal tract.

Here, we will focus on a specific category of pitch detection errors, the halving and doubling errors, in which the fundamental frequency F0 is confused with one of its harmonics (or sub-harmonics), generating incorrect assignments to multiple (or sub-multiple) frequency values of the correct one (Murray 2001). More precisely, F0 doubling errors occur when the estimated fundamental frequency is an overtone of the real fundamental frequency; on the other hand, F0 halving errors occur when the F0 determination algorithm erroneously mistakes the correctly estimated fundamental frequency by dividing the correct F0 value by some multiple of two. The most sophisticated algorithms tend to apply appropriate post-processing procedures in order to properly identify the correct value, among several possible candidates typically ranked in some way by the F0 extraction algorithm.

We will return to our brief description of halving and doubling errors later in this paragraph; now we provide a description of the smoothing method proposed by (Bagshaw 1994) in order to clarify the problem. The main purpose of this procedure is to distinguish between legitimate variations in the pitch profile and errors, trying to correct these in the best way. In particular, there is the assumption that F0 can grow between a frame and the next one to the maximum of the 75% and consequently it can drop to the 25% of the value of the first of the two frames. All values outside this range are considered respectively doubling and halving errors. At this point each voiced section of the utterance is processed separately: all the F0 values in the different frames which make up the voiced area are divided in various groups, each of them denoted by an index between -2 and 2. The partition begins putting the F0 values in the group identified with the index 0 as long as the transition among two subsequent frames generates a potential halving or doubling error. If this happens, the following F0 values are respectively positioned in the group identified with the index -1 or 1. The procedure continues in this way until all the values in the voiced region are placed in a group, changing the index of the group each time a potential error is detected. When the operation of subdivision of F0 values in each group ends, the procedure of correction of halving and doubling errors begins: the group containing the largest quantity of values is identified, defining it as the condition of normality (it could be the

group indexed with 0 or even a different group). Then, groups with a higher index are considered containing doubling errors while groups with a lower index are considered containing halving errors. Consequently the entire set of errors is corrected multiplying and dividing by powers of 2 the F0 values collected in the groups that identify incorrect estimates of the real fundamental frequency value.

We report also the research carried out by (Brøndsted 1997) according to which for a specific dialect of Danish, the presence of a glottal consonant "stød" can cause a pitch tracker to incorrectly report a halved value, as an example of a pitch tracking problem intimately connected with specific phonetic configurations. A further step would be to coordinate descriptions of pitch tracking doubling and halving errors with respect to categorizations of laryngealization (sometimes called creaky voice). This is a special kind of phonation in which the arytenoid cartilages in the larynx are drawn together; as a result, the vocal folds are compressed rather tightly, becoming relatively slack and compact. They normally vibrate irregularly at 20-50 pulses per second, about two octaves below the frequency of normal voicing, and the airflow through the glottis is very slow. Although creaky voice may occur with very low pitch, as at the end of a long intonation unit, it can also occur with a higher pitch (Titze 1994). The phenomenon of laryngealization is involved in the context of "cut-off" words, for example those words that a speaker does not complete (Shriberg 1999).

A better recognition of glottal pulses may lead to improve cut-off words recognition which are difficult phenomena to determine for a pitch tracker and consequently for an Automatic Speech Recognition (ASR) system too. Regarding this aspect, one can opt for an harmonic model of speech, which has gained considerable attention recently. This model takes into account the harmonic nature of voiced speech and it can be formulated to estimate pitch candidates with maximum likelihood criterion. Without entering deeply into the matter, the popular source-channel model of voiced speech considers glottal pulses as a source of period waveforms which is being modified by the shape of the mouth assumed to be a linear channel. Thus, the resulting speech is rich in harmonics of the glottal pulse period (Stylianou 1996). Like in other PDAs, pitch doubling and halving errors affect the harmonic model too; in order to solve these problems, one can opt for a local smoothing function that exploits the fact that there is more energy in the harmonics near the true pitch than at the corresponding neighbourhoods of half or double of its value. A local smoothing function is employed to include this energy and improve the strength of the pitch candidates in each frame. The harmonic model requires specification of the number of harmonics and the optimal choice depends on noise conditions (Asgari and Shafran 2013).

Here we provided a brief analysis of doubling and halving errors, a description of a procedure of pitch smoothing, some language dependent problems and the employment of the harmonic model to solve some of them. Starting from the next section we put our attention on our own proposal.

3. A Neural PDA smoother

The main purpose of our research work was an attempt to improve the performances of the Pitch Detection Algorithms. It is relevant to underline that all PDAs embody, as a last stage, some kind of smoothing algorithm trying to capture and correct mistakes in the intonation profile. As discussed before, these methods are often not sufficient to provide a reliable contour throughout the whole utterance. The Neural Smoother we are proposing tries to further improve profile smoothing applying more powerful techniques.

Our first assumption regarded the typology of the artificial neural network to employ. In order to correct the PDAs results, our pitch smoother needed to operate an increasingly precise approximation from the pitch input sequence to be improved to the gold standard output target obtained from the laryngograph. Having configured our problem as a sequence-to-sequence

mapping, we employed a particular architecture of recurrent neural network (RNN) suitable for this kind of problem.

These networks are recurrent because they perform the same computations for all the elements of a sequence of inputs, and the output of each element depends, in addition to the current input, from the previous state. RNNs have proved to have excellent performances in problems such as predicting the next character in a text or, similarly, the prediction of the next word in a sentence. They are also used for more complex problems, such as Machine Translation and Text Summarisation. In the former case, the network gets as input a sequence of words in a source language, while the output will be translated from the input sequence in a target language. Finally, other applications of great importance in which the RNNs are widely used are speech recognition and also image recognition.

A Long Short Term Memory (LSTM) neural net is a special Recurrent Neural Network architecture that was originally conceived by (Hochreiter and Schmidhuber 1997). This kind of neural network has gained a lot of attention in the context of deep learning because it offers excellent results and performances. The LSTM based networks are ideal for temporal sequences prediction and classification, replacing many traditional approaches to deep learning.

LSTM is a network composed by cells (LSTM blocks) linked to each other. Each LSTM block contains three types of gate: Input gate, Output gate, and Forget gate, which broadly implement, respectively, the function of writing, reading, and resetting on the cell memory. More precisely, the Input gate regulates the possibility for a new value to enter into the cell, the Forget gate determines if the value will be retained into the cell or not and the Output gate controls to which extent the cell value is transferred into the block output. Some of the connections between the LSTM elements are recurrent and all the weights of the connections have to be learned during the training process. The presence of these gates allows LSTM cells to remember information for a long time reducing the problem of the vanishing/exploding gradients during the training.

Mathematically, we can formalise the behaviour of a standard LSTM cell as

$$
\begin{aligned}
f_t &= \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \\
i_t &= \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \\
o_t &= \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \\
c_t &= f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \\
h_t &= o_t \circ \sigma_h(c_t)
\end{aligned}
\tag{1}
$$

where $x_t \in \mathbb{R}^d$ is input vector to the LSTM unit, $f_t \in \mathbb{R}^h$ the input gate's activation vector, $o_t \in \mathbb{R}^h$ the output gate's activation vector, $h_t \in \mathbb{R}^h$ the hidden state vector also known as output vector of the LSTM unit, $c_t \in \mathbb{R}^h$ the cell state vector, $W \in \mathbb{R}^{h \times d}, U \in \mathbb{R}^{h \times h}, b \in \mathbb{R}^h$ the weight matrices and bias vectors parameters which need to be learned during training, $\sigma_g, \sigma_h, \sigma_c$ the activation functions and the superscripts d and h refer to the number of input features and to the number of hidden units, respectively.

More specifically, in our case study we decided to employ a bidirectional LSTM. Bidirectional neural networks are based on the idea that the output at time t may depend on previous and future elements in the sequence. To realize this, the output of two neural networks must be mixed: one executes the process in one direction and the second in the opposite direction by processing the reversed input sequence. The network splits neurons of a normal recurrent neural network into two directions, one for positive time verse (forward states), and another for negative time verse (backward states) concatenating the outputs of the two networks. By this structure, the output

layer can get information from past and future states. We decided to opt for bidirectional LSTMs in order to have a better performance in our sequence learning (or approximation) problem.

We decided also to one-hot encode all the frames of our sequences, in order to obtain better performances in our sequence learning task. For our specific case, since we were working on female e male sources in both our datasets, we chose an interval of [0, 499] Hz for the number of features. Therefore, we transformed the F0 values determined for each frame in order to obtain input/output one-hot vectors; on the other hand, for the final evaluation of the predictions made by our model, we reversed this transformation getting common pitch values in the interval [0-499] Hz. This encoding of input and output data leads to input/output vectors of size 500 in our neural network model.

4. Experiments

4.1 Neural PDA setup

We implemented our pitch smoother in Python adopting Keras and Tensorflow. We defined a bi-directional Long Short Term Memory neural network layer with 100 neurons for one direction of the sequence and 100 neurons for the other direction, with a total of 200 LSTM units. A TimeDistributed layer has been wrapped around the output layer so that one value per timestep could be predicted given the full sequence provided as input. This allowed the LSTM hidden layer to return a sequence of values (one per timestep) rather than a single value for the whole input sequence. The network was optimised by using the categorical cross entropy loss function and the Adam optimiser algorithm (Kingma and Ba 2015).

4.2 Tested PDAs

We chose to test the three PDAs exhibiting the best performances in (Tamburini 2013), namely RAPT, SWIPE' and YAAPT. Even though they were originally developed as MATLAB functions, we decided to adopt the corresponding Python implementations and thus, as a first step, we have to test the correspondence of performances of the python implementations with the original ones in MATLAB.

4.2.1 A Robust Algorithm for Pitch Tracking (RAPT)

The primary purpose in the development of RAPT (Talkin 1995) was to obtain the most robust and accurate estimates possible, with little thought to computational complexity, memory requirements or inherent processing delay. This PDA was designed to work at any sampling frequency and frame rate over a wide range of possible F0, speaker and noise condition. In fact, although the delay inherent in RAPT probably disqualifies it from use in standard telephony, it does operate continuously and can be used anywhere. About this matter, several efficiency enhancements have been incorporated that significantly reduce computational costs while maintaining the desired accuracy. More specifically, for the determination of the pitch profile, RAPT adopts a Normalized Cross-Correlation Function (NCCF) and each candidate of F0 is estimated thanks to dynamic programming (also known as dynamic optimization, a method employed for solving a complex problem by breaking it down into a collection of simpler subproblems). The Python implementation we used is available at `http://sp-tk.sourceforge.net/`.

4.2.2 The Sawtooth Waveform Inspired Pitch Estimator (SWIPE/SWIPE')

SWIPE (Camacho 2007) improves the performance of pitch tracking adopting these measures: it avoids the use of the logarithm of the spectrum, it applies a monotonically decaying weight

to the harmonics, then the spectrum in the neighbourhood of the harmonics and middle points between harmonics are observed and smooth weighting functions are used. We will not focus on an overview of the mathematical expression of this PDA, but, in general, the algorithm can be described as the computation of the similarity between the square-root of the spectrum of the signal and the square-root of the spectrum of a sawtooth waveform, using a pitch dependent optimal window size. This definition gave rise to the name Sawtooth-Waveform Inspired Pitch Estimator (Camacho 2007). In our research we adopted SWIPE', a variant of this PDA that adopts only the main harmonics for pitch estimation, implemented in Python and available again at `http://sp-tk.sourceforge.net/`.

4.2.3 Yet Another Algorithm for Pitch Tracking (YAAPT)

YAAPT (Zahorian and Hu 2008) is a fundamental frequency (Pitch) tracking algorithm which was designed to be highly accurate and very robust for both high quality and telephone speech. One of the key features of YAAPT is the usage of spectral information to guide F0 tracking. Spectral F0 tracks can be derived by using the spectral peaks which occur at the fundamental frequency and its harmonics. It is experimentally shown that the F0 track obtained from the spectrogram is useful for refining the F0 candidates estimated from the acoustic waveform, especially in the case of noisy telephone speech (Zahorian and Hu 2008). With relation to the functioning of this PDA, a preprocessing step is employed to create multiple versions of the signal. Consequently, spectral harmonics correlation techniques (SHC) and a Normalized Cross-Correlation Function (as in RAPT) are adopted. The final profile of F0 is estimated thanks to dynamic programming techniques. For our experiments we employed pYAAPT, a Python implementation available at `http://bjbschmitt.github.io/AMFM_decompy/pYAAPT.html`.

4.3 Gold Standards

The evaluation tests were based on two English corpora considered as gold standards, both freely available and widely used in literature for the evaluation of PDAs:

- Keele Pitch Database - KPD[2] (Plante, Meyer, and Ainsworth 1995): it is composed of 10 speakers, 5 males and 5 females, who read, in a controlled environment, a small phonetically balanced text (the 'North Wind story'). The corpus contains also the output of a laryngograph, from which it is possible to accurately estimate the value of F0.
- FDA[3] (Bagshaw, Hiller, and Jack 1993): it is a small corpus containing 5' of recordings divided into 100 utterances, read by two speakers, a male and a female, particularly rich in fricative sound, nasal, liquid and glide, sounds particularly problematic to be analysed by the PDAs. Also in this case the gold standard for the values of F0 is estimated starting from the output of the laryngograph.

It is worth noticing that each of these datasets contains the output of a laryngograph. This instrument is composed of a pair of disc electrodes to record the vibrations around the throat. Electroglottography (EGG) signals record the time varying displacement of air particles at the glottis during the production of voiced sounds such as vowels, semi-vowels, nasals, diphthongs and voiced consonants. The electrodes are placed, non-invasively, at either side of the larynx.

2 `https://lost-contact.mit.edu/afs/nada.kth.se/dept/tmh/corpora/KeelePitchDB/`
3 `http://www.cstr.ed.ac.uk/research/projects/fda/`

A high-frequency electric current is applied, and due to variance in electrical impedance from the opening and closing of the glottis, an electroglottogram can be produced. There are several advantages of using EGG, the most significant being to reduce background noise. By eliminating irrelevant signals, EGG can increase the accuracy in the identification of perceived pitch. In the future, real-life applications of EGG can be developed due to its ability to reduce background noise, such as a wireless EGG integrated with clothes (Hui et al. 2015). This fact had crucial implications for the aims of our contribution: using KPD we encountered a few problems due to corrupted data. As (Plante, Meyer, and Ainsworth 1995) pointed out, where they knew that there was voiced speech but the larynx trace was corrupted, the data have been set to -1 (this happened sometimes because the measurements were based on two electrodes on the skin, which could lose contact as the speakers moved around). We will explain later how we decided to treat these corrupted data.

To perform our experiments, we had to split our datasets into a training set, a validation set and a test set. Consequently, we trained our model on the training set, we used the validation set to tune the hyperparameters of our smoother and finally the test dataset was employed to provide a balanced evaluation of our final model. This procedure was adopted both on KPD and FDA files, considering the output sequences of our PDAs and the gold standards obtained from the laryngograph. The main differences among the two datasets were the total number of files (10 speech samples for KPD and 100 for FDA) and the size of the files themselves. In fact, the original KPD files were much bigger than those of FDA, thus we decided to split each of them into 4 slices obtaining 40 speech samples. Considering that our purpose was trying to correct the sequences of the output of RAPT, pYAAPT and SWIPE' PDAs, we had in total 6 experiments (3 PDAs x 2 datasets).

In order to operate a significant subdivision between female and male files, we present our splitting for Keele Pitch Database:

	Training set	**Validation set**	**Test set**
Females	12	4	4
males	12	4	4

Here, instead, the splitting for FDA:

	Training set	**Validation set**	**Test set**
Females	34	8	8
males	34	8	8

We considered also the possibility of joining the two datasets in order to see if we get some improvements (Mixed configuration), and we followed the splitting

	Training set	**Validation set**	**Test set**
Females	46	12	12
males	46	12	12

All the splittings are speaker based as the speakers in the validation and test sets are not part of the training set.

4.4 Evaluation metrics

Proper evaluation mechanisms have to introduce suitable quantitative measures of performance that should be able to grasp the different critical aspects of the problem under examination. In (Rabiner et al. 1976) a de facto standard for PDA assessment measures is established, a standard used by many others after him (e.g. (Chu and Alwan 2009)). Given $E_{voi \to unv}$ and $E_{unv \to voi}$, respectively representing the number of frames erroneously classified between voiced

and unvoiced and vice versa, and E_{f0}, denoting the number of voiced frames in which the pitch value produced by the PDA differs from the gold standard for more than 16Hz, then we can define:

- Gross Pitch Error:

$$GPE = E_{f0}/N_{voi}$$

- Voiced Detection Error:

$$VDE = (E_{voi \to unv} + E_{unv \to voi})/N_{frame}$$

where N_{voi} is the number of voiced frames in the gold standard and N_{frame} is the number of frames in the utterance. These indicators, taken individually or in pairs, have been used in a large number of works to evaluate the performance of PDAs. The two indicators, however, measure very different errors; it is possible to measure the performance using only one indicator, usually GPE, but it evaluates only part of the problem and hardly provide a faithful picture of PDA behaviour. On the other hand, considering both measures leads to a difficult comparison of the results.

In order to find a remedy to these problems, (Lee and Ellis 2012) suggested slightly different metrics, which allow the definition of a single indicator:

- Voiced Error:

$$VE = (E_{f0} + E_{voi \to unv})/N_{voi}$$

- Unvoiced Error:

$$UE = E_{unv \to voi}/N_{unv}$$

- Pitch Tracking Error:

$$PTE = (VE + UE)/2$$

where N_{unv} is the number of unvoiced frames contained in the gold standard. However, trying to interpret the results obtained by a PDA in light of the PTE measurement is rather complex: it is not immediate to identify from the obtained results the most relevant source of errors.

In light of what has been said previously, it seems appropriate to introduce a new measure of performance that is able to easily capture the performance of a PDA in a single, clear indicator that considers all types of possible errors to be equally relevant. So, following (Tamburini 2013), we adopted the Pitch Error Rate as performance metric, defined as:

$$PER = (E_{f0} + E_{voi \to unv} + E_{unv \to voi})/N_{frame}$$

This measure sum all the types of possible errors without privileging or reducing the contribution of any component and allowing a simpler interpretation of the obtained outcomes.

5. Results

5.1 Preliminary Evaluation

We repeated the same experiments as in (Tamburini 2013) with the Python implementations of the chosen algorithms in order to check the employed codes and to derive common baselines.

Obviously a few small differences in performances will be encountered. Table 1 shows the performance values obtained by the three algorithms compared to all the measures considered for both the gold standards used in the evaluation. We consider these results as baseline performance.

Table 1
The experiments in Tamburini (2013) reproduced using the considered PDA python implementations.

Keele Pitch Database						
PDA	PER	GPE	VDE	PTE	VE	UE
pYAAPT	0.14056	0.05517	0.09777	0.09433	**0.1132**	0.07539
RAPT	**0.12596**	0.04917	**0.08806**	**0.08498**	0.11966	**0.05031**
SWIPE'	0.14236	**0.03556**	0.11474	0.09623	0.12867	0.0638
FDA Corpus						
PDA	PER	GPE	VDE	PTE	VE	UE
pYAAPT	0.11912	0.05381	0.08889	0.08689	0.11016	0.06361
RAPT	**0.09533**	0.03591	**0.07554**	**0.07159**	**0.09637**	**0.0468**
SWIPE'	0.10594	**0.02543**	0.09208	0.07863	0.10652	0.05074

The performances obtained for the FDA corpus are generally better; maybe the algorithms suffer the length of the speech files. As we pointed above, in fact, KPD is a larger corpus with definitely bigger files even if we splitted each of them into four slices. Another important consideration that has to be made, regards the corrupted data in the KPD: removing them from the sequences probably got worse the final evaluation, affecting the total length of the sequences themselves. Furthermore, it has to be kept in mind that we used Python implementations of these algorithms that, as we pointed out some times earlier, are originally available as MATLAB functions. We do not have the proof that this implementation difference affects the results, but more work about checking this issue should be done in the future. Leaving aside these considerations, let us focus on the performances. It can be observed easily that RAPT reaches the best achievements both on KPD and FDA corpus. In evaluating the results obtained, it seems appropriate to study more accurately the types of errors that the three algorithms exhibited in the automatic detection of F0; Table 2 focuses on the total Pitch Error Rate and how this is distributed with respect to the three types of errors that make up its definition, namely E_{f0}, $E_{voi \to unv}$, $E_{unv \to voi}$.

Table 2 shows quite different behaviours among the three pitch detection algorithms: the errors committed seem to be distributed among the different types of error in an uneven way and with different configurations between the PDAs. It could therefore be useful to consider the possibility of combining the contributions of the different algorithms as an attempt to improve their performances. One possibility to do this was to consider, as an estimate of the pitch value in a certain frame, the median of the values calculated by an odd number of different algorithms (in this specific case study, three different PDAs) as it has been done by (Tamburini 2013).

Table 2
Error analysis on the experiments in Tamburini (2013) reproduced using the considered PDA python implementation. We added a further algorithm 'Median', proposed in the cited study, that, for each frame, keeps the median value among the three F0 values proposed by the considered PDAs.

Keele Pitch Database				
PDA	PER	E_{f0}	$E_{voi \to unv}$	$E_{unv \to voi}$
pYAAPT	0.14056	0.04278	0.04411	0.05366
RAPT	0.12596	0.03789	0.05252	**0.03554**
SWIPE'	0.14236	0.02762	0.06985	0.04488
Median	**0.08814**	**0.02656**	**0.03359**	0.03564
FDA Corpus				
PDA	PER	E_{f0}	$E_{voi \to unv}$	$E_{unv \to voi}$
pYAAPT	0.11912	0.03023	**0.03399**	0.0549
RAPT	0.09533	0.01978	0.03438	0.04116
SWIPE'	0.10594	**0.01385**	0.04773	0.04434
Median	**0.10182**	0.02537	0.03686	**0.03917**

From Table 2 it emerges quite clearly how the combination of different algorithms with the median method makes better results. In particular, it is worth underlying how much the E_{f0} error decreases, especially in the experiments involving KPD.

This section presented an objective evaluation of three algorithms for the automatic extraction of the fundamental frequency value in the spoken language, using a large set of different metrics. It will be useful as a baseline for comparing the performances of the proposed neural PDA smoother.

5.2 Neural PDA Evaluation

In order to carry out an objective evaluation of our pitch smoother, we decided to put our attention on one of the metrics employed for the evaluation of the three Pitch Detection Algorithms, namely the Pitch Error Rate (PER). In fact, as we pointed out earlier, this measure is able to easily capture the performance of a PDA in a single, clear indicator that considers all types of possible errors to be equally relevant.

After the influential paper from (Reimers and Gurevych 2017) it is clear to the community that reporting a single score for each DNN training session could be heavily affected by the system initialisation point and we should instead report the mean and standard deviation of various runs with the same setting in order to get a more accurate picture of the real systems performances and make more reliable comparisons between them.

The PER metric was computed for each epoch during the training phase for all subsets in order to determine the stopping epoch when we get the minimum PER on the validation set. We performed 10 runs for each experiment computing means, standard deviations and significance tests.

We also tested our pitch smoother on the mixed configurations of the datasets employed, adopting the same procedures.

Table 3 shows all the obtained results. The proposed system always exhibits the best results in any experiment with relevant performance gains with respect to the PDAs base outputs. All the differences resulted highly significant when applying a t-test. Given the very small standard

deviation in all the experiments we can conclude that, in this case, the initialisation point did not affect the neural network performances too much.

Table 3
PER mean (μ) and standard deviation (σ) obtained by the proposed pitch profile smoother. One sample t-test significance test returns p$\ll$0.001 for all experiments. N.B.: Even if the number of experiments is small (10), the power analysis of the t-tests is always equal to 1.0 showing maximum t-test reliability. The assumption of normality has been tested, with the Shapiro-Wilk test, before computing the t-test.

Keele Pitch Database			
PDA	PDA PER	Smoother PER μ	Smoother PER σ
pYAAPT	0.14056	**0.07958**	0.00271
RAPT	0.12596	**0.08481**	0.00376
SWIPE'	0.14236	**0.10065**	0.00292
FDA Corpus			
PDA	PDA PER	Smoother PER μ	Smoother PER σ
pYAAPT	0.11912	**0.06731**	0.00421
RAPT	0.09533	**0.06752**	0.00232
SWIPE'	0.10594	**0.07769**	0.00212
Mixed Keele+FDA Corpus			
PDA	PDA PER	Smoother PER μ	Smoother PER σ
pYAAPT	0.06951	**0.06302**	0.00246
RAPT	0.09859	**0.07256**	0.00297
SWIPE'	0.08758	**0.08151**	0.00144

Referring to the performance outcomes of the Pitch Detection Algorithms we provided in Table 3, it can be easily noted a general, great improvement. For both the configurations we employed, pYAAPT shows the best performances; the category in which we observe the bigger error in each of our combinations is $E_{voi \to unv}$, the number of frames erroneously classified between voiced and unvoiced; this means that our smoother has a major struggle in correctly identifying the boundaries between voiced and unvoiced regions. Despite this, our pitch smoother behaves rather well in correcting all halving and doubling errors, which are collected in E_{f0}, the indicator that measures the error of estimation of the F0 values on frames considered voiced.

We performed a one sample t-test significance test that returned $p \ll 0.001$ for all experiments and, even if the number of experiments is small (10), the power analysis of the t-tests was always equal to 1.0, showing maximum t-test reliability.

6. Conclusions

This paper presented a new pitch smoother based on recurrent neural networks that obtained excellent results when evaluated using two standard benchmarks for English. The results showed that our smoother is able to efficiently learn how to smooth a pitch profile produced by widely used PDAs removing halving and doubling errors from the profile. The proposed Neural

Smoother will not increase the total processing time for each utterance as, once properly trained, is able to process and correct a single intonation profile very quickly.

Future works could regard the intermixing of various corpora in different languages in order to test the possibility of deriving a pitch smoother able to properly work without caring about language and, possibly, specific corpora and language registers. In principle we can imagine that it would be possible to train a neural pitch smoother like the one presented in this paper cross-linguistically to correct the pitch detection errors and apply it to smooth the PDAs profiles obtained on different languages and registers. This is a pure speculation and we definitively have to perform new experiments in order to verify this idea. The main problem in performing such experiments is the availability of speech corpora provided with the laryngograph profiles. We need definitely a good sample of, at least, different languages to perform these experiments and, at the time of writing, we have only few corpora of this kind.

Acknowledgements

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

References

Asgari, Meysam and Izhak Shafran. 2013. Improving the accuracy and the robustness of harmonic model for pitch estimation. In *Proceedings of 13th Annual Conference of the International Speech Communication Association - Interspeech 2013*, pages 1936–1940, Lyon, France, August.

Babacan, Onur, Thomas Drugman, Nicolas D'Alessandro, Nathalie Henrich, and Thierry Dutoit. 2013. A comparative study of pitch extraction algorithms on a large variety of singing sounds. In *IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP 2013*, pages 7815–7819, Vancouver, Canada, May.

Bagshaw, Paul C. 1994. *Automatic prosodic analysis for computer-aided pronunciation teaching*. Ph.D. thesis, University of Edimburgh.

Bagshaw, Paul C., Steven M. Hiller, and Mervyn A. Jack. 1993. Enhanced pitch tracking and the processing of f0 contours for computer aided intonation teaching. In *Proceedings of Eurospeech '93*, pages 1003–1006, Berlin, September.

Bartosek, Jan. 2010. Pitch detection algorithm evaluation framework. In *Proceedings of 20th Czech-German Workshop on Speech Processing*, pages 118–123. Prague, Czech Republic, September.

Brøndsted, Tom. 1997. Intonation contours 'distorted' by tone patterns of stress groups and word accent. In *Intonation: Theory, Models and Applications: Proceedings of an ESCA Workshop*, pages 55–58, Athens, Greece, September.

Byrd, Dani and Toben H.Mintz. 2010. *Discovering Speech, Words, and Mind*. Wiley-Blackwell.

Camacho, Arturo. 2007. *SWIPE: A sawtooth waveform inspired pitch estimator for speech and music*. Ph.D. thesis, University of Florida.

Chu, Wei and Abeer Alwan. 2009. Reducing f0 frame error of f0 tracking algorithms under noisy conditions with an unvoiced/voiced classification frontend. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP 2009*, pages 3969–3972, Taipei, Taiwan, April.

Chu, Wei and Abeer Alwan. 2012. Safe: A statistical approach to f0 estimation under clean and noisy conditions. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(3):933–944.

de Cheveigné, Alain and Hideki Kawahara. 2002. Yin, a fundamental frequency estimator for speech and music. *Journal of the Acoustical Society of America*, 111(4):1917–1930.

Gawlik, Mateusz and Wieslaw Wszolek. 2018. Modern pitch detection methods in singing voices analyzes. In *Proceedings of Euronoise 2018*, pages 247–253, Crete, May.

Gonzalez, Sira and Mike Brookes. 2014. PEFAC-A pitch estimation algorithm robust to high levels of noise. *IEEE Transactions on Audio, Speech, and Language Processing*, 22(2):518–530.

Han, Kun and DeLiang Wang. 2014. Neural network based pitch tracking in very noisy speech. *IEEE Transactions on Audio, Speech, and Language Processing*, 22(12):2158–2168.

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

Huang, Feng and Tan Lee. 2012. Robust pitch estimation using l1-regularized maximum likelihood estimation. In *Proceedings of 13th Annual Conference of the International Speech Communication Association - Interspeech 2012*, pages 378–381, Portland (OR), September.

Hui, Lu, Lu Hui Ting, Swee Lan See, and Paul Y. Chan. 2015. Use of electroglottograph (EGG) to find a relationship between pitch, emotion and personality. *Procedia Manufacturing*, 3:1926–1931.

Jang, Seung-Jin, Seong-Hee Choi, Hyo-Min Kim, Hong-Shik Choi, and Young-Ro Yoon. 2007. Evaluation of performance of several established pitch detection algorithms in pathological voices. In *Proceedings of the International Conference of the IEEE Engineering in Medicine and Biology Society - EMBC 2007*, pages 620–623, Lyon, France, August.

Jin, Zhaozhang and DeLiang Wang. 2011. Hmm-based multipitch tracking for noisy and reverberant speech. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(5):1091–1102.

Jlassi, Wided, Aicha Bouzid, and Noureddine Ellouze. 2016. A new method for pitch smoothing. In *2nd International Conference on Advanced Technologies for Signal and Image Processing*, pages 657–661, Monastir, Tunisia, March.

Jouvet, Denis and Yves Laprie. 2017. Performance analysis of several pitch detection algorithms on simulated and real noisy speech data. In *25th European Signal Processing Conference - EUSIPCO 2017*, pages 1614–1618, Kos, Greece, September.

Kellman, Michael R. and Nelson Morgan. 2017. Robust multi-pitch tracking: a trained classifier based approach. Technical report, ICSI Tchnical Report, Berkeley, CA.

Kingma, Diederik P. and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations - ICLR 2015*, San Diego, CA, May.

Kotnik, Bojan, Harald Höge, and Zdravko Kacic. 2006. Evaluation of pitch detection algorithms in adverse conditions. In *Proceedings of Speech Prosody 2006*, PS2-8-83, Dresden, May.

Lee, Byung Suk and Daniel P. W. Ellis. 2012. Noise robust pitch tracking by subband autocorrelation classification. In *Proceedings of 13th Annual Conference of the International Speech Communication Association - Interspeech 2012*, pages 707–710, Portland (OR), September.

Luengo, Iker, Ibon Saratxaga, Eva Navas, Inmaculada Hernaez, Jon Sanchez, and Inaki Sainz. 2007. Evaluation of pitch detection algorithm under real conditions. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP 2007*, pages 1057–1060, Honolulu, Hawaii, April.

Murray, Kathleen. 2001. A study of automatic pitch tracker doubling/halving errors. In *Proceedings of 2nd SIGdial Workshop on Discourse and Dialogue*, Aalborg, Denmark, September.

Plante, Fabrice, Georg F. Meyer, and William A. Ainsworth. 1995. A pitch extraction reference database. In *Proceedings of Eurospeech '95*, pages 837–840, Madrid, September.

Rabiner, Lawrence R., Michael J. Cheng, Aaron E. Rosenberg, and Carol A. McGonegal. 1976. A comparative performance study of several pitch detection algorithms. *IEEE Transaction on Acoustics, Speech and Signal Processing*, 24(5):399–418.

Reimers, Nils and Iryna Gurevych. 2017. Reporting score distributions makes a difference: Performance study of lstm-networks for sequence tagging. In *Proceedings of Conference on Empirical Methods in Natural Language Processing - EMNLP 2017*, pages 338–348, Copenhagen, Denmark, September.

Shriberg, Elizabeth E. 1999. Phonetic consequences of speech disfluency. In *Proceeedings of the International Congress of Phonetic Sciences - ICPhS '99*, pages 619–62, San Francisco, August.

So, YongJin, Jia Jia, and LianHong Cai. 2012. Analysis and improvement of auto-correlation pitch extraction algorithm based on candidate set. In Q. Zhihong, C. Lei, S. Weilian, W. Tingkai, and Y. Huamin, editors, *Recent Advances in Computer Science and Information Engineering: Volume 5*. Springer, Heidelberg/Berlin, pages 697–702.

Stylianou, Yannis. 1996. *Harmonic plus noise models for speech, combined with statistical methods, for speech and speaker modification*. Ph.D. thesis, Ecole Nationale Superieure des Telecommunications.

Sukhostat, Lyudmila and Yadigar Imamverdiyev. 2015. A comparative analysis of pitch detection methods under the influence of different noise conditions. *Journal of Voice*, 29(4):410–417.

Talkin, David. 1995. A robust algorithm for pitch tracking (RAPT). In W. B. Kleijn and K. K. Paliwal, editors, *Speech Coding and Synthesis*. Elsevier, New York, pages 495–518.

Tamburini, Fabio. 2013. Una valutazione oggettiva dei metodi più diffusi per l'estrazione automatica della frequenza fondamentale. In *Atti dell IX Convegno Nazionale dell'Associazione Italiana di Scienze della Voce - AISV 2013*, pages 427–434, Roma, January.

Titze, Ingo R. 1994. *Principles of Voice Production*. Prentice Hall.

Veprek, Peter and Michael S. Scordilis. 2002. Analysis, enhancement and evaluation of five pitch determination techniques. *Speech Communication*, 37(3-4):249–270.

Wang, Dongmei and Philipos C. Loizou. 2012. Pitch estimation based on long frame harmonic model and short frame average correlation coefficient. In *Proceedings of 13th Annual Conference of the International Speech Communication Association - Interspeech 2012*, pages 923–926, Portland, OR, September.

Wu, Mingyang, DeLiang Wang, and Guy J. Brown. 2003. A multipitch tracking algorithm for noisy speech. *IEEE Transactions on Audio, Speech, and Language Processing*, 11(3):229–241.

Zahorian, Stephen A. and Hongbing Hu. 2008. A spectral/temporal method for robust fundamental frequency tracking. *Journal of the Acoustical Society of America*, 123(6):4559–4571.

Zhao, Xufang, Douglas O'Shaughnessy, and Nguyen Minh-Quang. 2007. A processing method for pitch smoothing based on autocorrelation and cepstral f0 detection approaches. In *Proceedings of the International Symposium on Signals, Systems and Electronics*, pages 59–62, Montreal, Canada, August.

Large scale datasets for Image and Video Captioning in Italian

Scaiella Antonio*
Università di Roma, Tor Vergata

Danilo Croce**
Università di Roma, Tor Vergata

Roberto Basili†
Università di Roma, Tor Vergata

The application of Attention-based Deep Neural architectures to the automatic captioning of images and videos is enabling the development of increasingly performing systems. Unfortunately, while image processing is language independent, this does not hold for caption generation. Training such architectures requires the availability of (possibly large-scale) language specific resources, which are not available for many languages, such as Italian.

In this paper, we present MSCOCO-it e MSR-VTT-it, two large-scale resources for image and video captioning. They have been derived by applying automatic machine translation to existing resources. Even though this approach is naive and exposed to the gathering of noisy information (depending on the quality of the automatic translator), we experimentally show that robust deep learning is enabled, rather tolerant with respect to such noise. In particular, we improve the state-of-the-art results with respect to image captioning in Italian. Moreover, in the paper we discuss the training of a system that, at the best of our knowledge, is the first video captioning system in Italian.

1. Introduction

Given the massive production of images and videos available from Social Networks and Distributed Sensors, automating the annotation, retrieval and clustering of the corresponding multimedia material is becoming crucial. Even though neural embeddings are growingly adopted to represent multimedia objects, linguistic descriptions also represent a straightforward, and more intuitive, representation of their contents. In fact, captions offer a simple way to summarize, index and search those contents implicit in such different types of data.

In this scenario, the goal of the automatic captioning of images and videos is thus to predict the correct caption(s) given an image or a video, respectively. In other words, a multimedia "captioner" is expected to automatically generate a textual description of a multimedia content, summarizing the depicted entities, the involved actions and those relations holding between them.

* Department of Enterprise Engineering - Via del Politecnico 1, 00133 Rome, Italy.
E-mail: scaiellantonio@gmail.com
** Department of Enterprise Engineering - Via del Politecnico 1, 00133 Rome, Italy.
E-mail: croce@info.uniroma2.it
† Department of Enterprise Engineering - Via del Politecnico 1, 00133 Rome, Italy.
E-mail: basili@info.uniroma2.it

Slightly more formally, given an image X as input, the output of an image captioner is $S(X) = \left(S_{\{1\}}, \ldots, S_{\{m\}}\right)$ such that $S(X)$ is a meaningful sentence where every $s_{\{i\}}$ is a word belonging to a vocabulary $V = \left(v_{\{1\}}, \ldots, v_{\{n\}}\right)$ of a given language. Similarly, considering the video as a sequence of images (frames), the output of a video captioner is again a meaningful sentence with the same characteristics.

Many recently proposed methods are based on deep neural networks. The results on the task show performances whose quality is sometimes comparable with humans judgments (Hossain et al. 2019). Some approaches directly operate on input multimedia sources, while in some works these are also contextualized within associated texts, i.e. (Feng and Lapata 2013; Batra, He, and Vogiatzis 2018). Most of the existing neural approaches are inspired by the architecture proposed in (Vinyals et al. 2015) where images are first encoded by a Convolutional Neural Network (which transforms them into continuous representations) and then "translated" into descriptive sentences by a recurrent architecture (for example, a Long Short-Term Memory network). At the same time, several approaches proposed for video captioning extend existing approaches for image captioning, while modeling a video as a sequence of images, such as (Venugopalan et al. 2015), or adopting more strict "Sequence to Sequence" approaches (Sutskever, Vinyals, and Le 2014; Yao et al. 2015).

In any case, the training of these neural architectures requires large scale collections of multimedia content paired with one or more captions. Important (and costly) effort led to the production of several datasets, most of which exist only for English. Examples are the MS-COCO dataset (Lin et al. 2014) for image captioning, made of 600,000 captions for about 120,000 images and the MSR-VTT dataset (Xu et al. 2016) for video captioning, made of 200,000 captions for 10,000 videos. Figure 1 reports an example from the MS-COCO dataset, i.e. an image with the corresponding five captions.

Figure 1
An example of image-caption from MSCOCO dataset

Other similar but smaller corpora exist for different languages, most of all composed of sentences that are manually annotated or translated in different languages:

IAPR-TC12, with 20,000 English, German and Spanish described images (Escalante et al. 2010), the Pascal Sentences Dataset, made of of 1,000 Japanese/English described images (Funaki and Nakayama 2015) and Multi30K (Elliott et al. 2016) made of 30,000 German/English described images.

In (Masotti, Croce, and Basili 2017, 2018), a possible alternative to the manual construction of such datasets is explored. The authors propose automatic machine translation as a way to derive annotated data through the direct translation of the original (English) material. In principle, the result is a large scale set of (images,captions) whose texts are in Italian and directly applicable to the training stage of a Neural architecture. Most noticeably, the work in (Masotti, Croce, and Basili 2018) empirically demonstrates that captions produced in Italian by neural models trained over the noisy dataset are of a better quality than the ones obtained by direct translations of English captions. In other words, the pairing of an automatic captioner with a translation system (both trained on manual annotations) is subject to a stronger performance drop than compared to the standard neural architecture trained over automatically translated input material.

In this paper, we thus propose two large scale resources to train neural architectures for image and video captioning in Italian[1]. They are derived by automatically translating the textual descriptions of images from MS-COCO and video from MSR-VTT. While the latter represents a brand new resource made of 200,000 video/caption pairs, the former dataset is generated by re-translating the original MS-COCO. Although this may seem redundant, we assume that the general improvements obtained in Automatic Translation since (Masotti, Croce, and Basili 2017), especially since the introduction of Trasformer-based architectures (Vaswani et al. 2017) will positively impact on the quality of derived neural captioners. In addition, we created a realistic test set as two sets of manually validated portions of both datasets: in fact, while model generation should be robust to noise in training material, representative performance measures strictly require validated material. We also investigate the performance of an Image Captioning model based on the Attention Mechanism (Xu et al. 2015) showing how the use of the new dataset instead of the old one, with the same model, improves the result. Finally, in parallel, we discuss the design and evaluation of the first neural system for Video Captioning in Italian, still based on Attention mechanisms.

In the rest of the paper, Section 2 introduces the resources developed in this work. In Section 3 the experimental evaluation of two neural architectures trained on these resources is discussed, while Section 4 derives the conclusions.

2. The corpus

In this section we present the two large-scale datasets for image and video captioning in Italian. These are obtained by automatic translation of the corresponding English versions[2]. It is worth noting that a subset of each corpus has been manually validated, in order to guarantee the sound evaluation of systems trained on possibly noisy annotated captions. Even though this is not the main focus of this work, this validated material also enables the evaluation of the automatic translation system, that obviously impacts

1 We publicly released both resources at the following GitHub links:
mscoco-it: `https://github.com/crux82/mscoco-it`
msr-vtt-it: `https://github.com/crux82/msr-vtt-it`

2 Captions have been translated by using Microsoft Azure Translator (`https://azure.microsoft.com/it-it/services/cognitive-services/translator-text-api/`) between July 2019 and August 2019.

the overall process: the higher the quality of the produced translations, the higher is the expected quality of the neural captioning system[3]. We thus compared the validated sentences with the automatic translations by using the `sacrebleu`[4] library, obtaining a BLEU score of 0.70 (Papineni et al. 2002). This score is very high, especially if compared with the traditional evaluations of modern translations systems: however, it must be said that our setting is easier if compared with standard machine translation ones. Here annotators are not asked to write the translations without knowing the output of the system, but they are asked to fix the produced translations. As a consequence, a higher number of common sub-sequences between the input sentences and the validated ones are expected, resulting in a higher BLUE score. In any case, this BLEU score suggests that the Italian material is characterized by a low level of noise due to the automatic translation process and it bodes well for the final quality of the captioning system.

2.1 Image Captioning

The image captioning task requires a large number of training examples and among existing datasets (Hossain et al. 2019), one of the largest one is MSCOCO (Lin et al. 2014). It was released in its first version in the 2014 and is composed approximately of 122,000 annotated images for training and validation, plus 40,000 more for testing. As shown in Fig. 1, each image is paired with 5 or 6 human-validated descriptions, for a total of 600k (image,caption) pairs fully available for the training and validation stages.

In particular, the original MSCOCO split consists in 82,783 captions composing the training dataset, 40,504 composing the validation set and 40,775 composing the test set. Unfortunately, captions in the test dataset are not publicly available, as they are only used in competitions. To overcome this issue, some works apply alternative splits. For example, the neural architecture proposed in (Vinyals et al. 2015) is trained on all the MSCOCO training set plus 85% of the validation set (approximately 116,000 training images, for a total of 580,000 training image-caption pair); 6,000 images from the validation set are left out and split in a development set and a test set of 2,000 and 4,000 images, respectively.

In Italian, the first version of MSCOCO-it (Masotti, Croce, and Basili 2017) follows the same specifications. A subset of captions from the original development set was manually validated (noted by v. in Table 1), thus resulting into 308 images as the development and 596 as the test set. Some (few) images were associated with captions that are only partially validated by annotators (denoted by p.). All the others, denoted by n., are left not analyzed. Overall, the statistics about the Italian dataset are shown, in terms of numbers of represented images and captions, together with the size of the resulting dataset as number of different tokens.

This work proposes a second version of MSCOCO-it where all training set plus an 85% of the validation set was fully re-translated. Here, we maintained the original validated translation, but also accomplish the validation for all the partially validated images. Validations were carried out by six annotators, not expert in Deep Learning or Natural Language Processing, but native Italian speakers.

Given the limited average length of input captions, (i.e. 10 words for caption) translations are of a good quality. For example: "*a man in shorts gets ready to hit a tennis*

3 Even though this score is measured only on the test datasets, we can speculate it reflects the quality of the translations also in the training/development subsets, since no bias is applied on this splitting.

4 `https://github.com/mjpost/sacreBLEU`

ball" is translated into "*un uomo in pantaloncini si prepara a colpire una palla da tennis*" or "*A group of three people standing on top of a snow covered slope*" into "*un gruppo di tre persone in piedi sulla cima di un pendio coperto di neve*". In some texts, word senses are mistakenly assigned, such as "*Three computer monitors sitting on top of a wooden table*" translated in "*Tre monitor per computer seduto sulla cima di un tavolo di legno*" or "*A vase of freshly cut flowers on a table*" into "*Un vaso di fiori freschi su una tabella*". In other cases, the translation is grossly incorrect, such as "*Man in body suit surfing on a large wave*" translated into "*Uomo nel vestito del corpo surf su un'onda di grandi dimensioni*" or "*a couple of kids are holding up umbrellas*" into '*Un paio di ragazzi sono holding up ombrelloni*". This is more common when jargon expressions (such as *"body suit"*) or informal expressions (e.g. not so common phrasal verbs) are employed in captions.

Table 1
Statistics for the MSCOCO-it dataset.

		#images	#captions	#words
training	n.	116,195	581,286	∼6,900,000
development	v.	308	1,516	∼18,000
	p.	(14)	25	∼300
	n.	1,696	8,486	∼102,000
test	v.	596	2,941	∼34,600
	p.	(23)	41	∼500
	n.	3,422	17,120	∼202,000

Table 2
Statistics for the second version MSCOCO-it-v2 of the MSCOCO-it dataset.

		#images	#captions	#words
training	n.	116,195	581,286	∼6,900,000
development	v.	308	1,541	∼18,000
	n.	1,696	8,486	∼102,000
test	v.	596	2,982	∼35,000
	n.	3,422	17,120	∼202,000

We are interested in evaluating the potential good impact of the novel resource in the neural training of the image captioner. The experimental results reported in the following sections will in fact connect the quality of the training material to the quality of an image captioner, which is trained over the two different dataset and compared on the same test set. Overall, it is worth noting the size of this (possible noisy) dataset made of hundred thousands of examples for a language (Italian) for which such resource has never been available.

2.2 Video Captioning

As for image captioning, several English benchmarks exist for Video Captioning. Examples of such datasets are MSVD, YouCook, M-VAD, TACoS, and MPII-MD (Aafaq et al. 2020). The first large-scale video benchmark for video understanding was MSR-VTT (Xu et al. 2016). In its current version (2017), MSR-VTT provides 10,000 web video clips

with 41.2 hours and 200,000 clip-sentence pairs. Each clip is annotated with about 20 natural sentences written by human annotators. This corpus is one of the largest open-domain video captioning datasets with a wide variety of video topics. In fact, the videos generically cover a comprehensive list of 20 categories (or topics), such as music, movie, cooking or sports. Table 3 shows the statistics of MSR-VTT dataset.

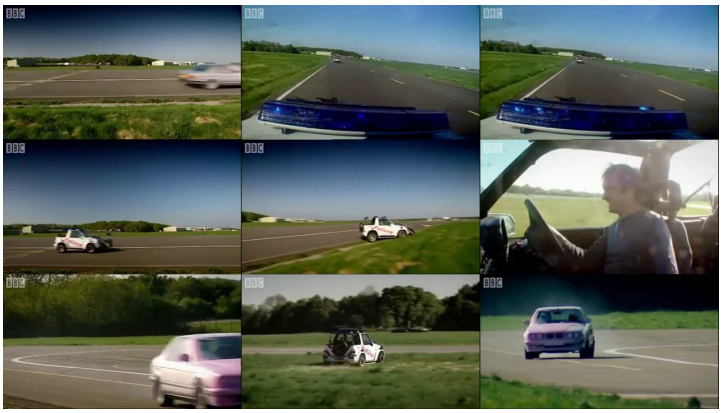


Figure 2
Example of a video included in the MSR-VTT dataset. One of the available captions for this video is *"a man is driving a small police car on a track"*.

By translating this dataset we obtained the first resource for the training of data-driven video captioning systems in Italian: MSR-VTT-it. The resource have the following video split: 6,513 video (and the corresponding captions) for training, 497 for validation and 2,990 for tests as summarized in Table 4. It is natural, like in the captioning task over MSCOCO-it, that some captions are not properly translated. The original captions of the video whose frames are shown in Figure 2 are the following:

1. *"a man is driving a small police car on a track"*
2. *"a british guy rides a police car through a grassy field"*
3. *"a man with a blue visored helmet is driving a car"*
4. *"there is a man driving a car into the grass"*
5. *"a car race is organized and displayed between three vehicles of vastly different performance"*

The translated captions are hereafter reported where wrong lexical choices or grammatical errors are underlined:

1. *"un uomo sta guidando una piccola auto della polizia su una pista"*
2. *"un ragazzo britannico cavalca una macchina della polizia attraverso un campo erboso"*
3. *"un uomo con un casco blu con visiera sta guidando una macchina"*
4. *"c'è un uomo alla guida di una macchina in erba"*
5. *"una gara automobilistica è organizzata ed esposta tra tre veicoli di prestazioni molto diverse"*

As for image captioning data, we developed a manually validated testset from a randomly selected set of 100 test videos, made thus of 2,000 validated images-caption pairs, reported in Table 4.

Table 3
MSR-VTT general statistics

#Video	7,180
#Clip	10,000
#Sentence	200,000
#Word	∼1,850,000
Vocabulary	29,316
Duration(hr)	41.1

The validation of the test set was carried out by six annotators which were asked only to check and correct the translations after watching the original video. Annotators are not expert in the field of Deep Learning or Natural Language Processing, but are Italian native speakers.

3. First evaluation

In this section we report the experimental evaluation of two different captioning systems enabled by the two resources presented in this work. In both cases we adopted an open source implementation of a deep architecture for image and video captioning, with the aim of maximizing the reproducibility of the obtained results. Systems were trained on a NVIDIA Tesla T4 GPU and evaluated using traditional metrics, i.e., BLEU (Papineni et al. 2002), METEOR (Lavie and Agarwal 2007), ROUGE (Lin 2004), Cider (Vedantam, Zitnick, and Parikh 2015).

Table 4
MSR-VTT-it statistics as the numbers of available video and Italian captions.

		#videos	#captions
training	n.	6,513	130,260
development	n.	497	9,940
test	v.	100	2,000
	n.	2,990	59,800

3.1 Image Captioning

The results reported in (Masotti, Croce, and Basili 2017) were obtained by adopting the architecture presented in (Vinyals et al. 2015) trained over a subset of the 20% of the training material. In this work we improved that evaluation in two directions. First, we trained a different architecture, based on the approach presented in (Xu et al. 2015), which exploits Attention Mechanisms[5]: these are in fact demonstrated to improve

5 We used the architecture implemented using PyTorch and available at the following link: `http://github.com/sgrvinod/a-PyTorch-Tutorial-to-Image-Captioning`

the quality of the generated captions since the architecture focuses on specific areas of the input images when generating each word. Second, the adoption of GPU-based hardware allowed to scale to the size of the entire training set.

Table 5
Italian Image captioning results

Model	Bleu_4	Cider	Rouge_L
(Masotti, Croce, and Basili 2018)	0.26	0.79	/
This work (old data)	0.28	0.93	0.48
This work (new data)	0.29	0.96	0.48

In a nutshell, the architecture combines a CNN, based on ResNet (He et al. 2016) which encodes the images into low-dimensional embeddings. Then a long short-term memory network produces the caption by generating one word at each time step, conditioned onto a context vector (which implements the Attention), the previous hidden state and the previously generated word. the context vector allows the LSTM, at each time step, to focus more carefully on some portions of the image rather than on all visual aspects by fostering a more modular learning of visual and lexical correlations.

To select the best network parameters, a validation was carried out over the development set by selecting those configuration achieving on average the best score on all the metrics we considered. The learning rate was set at standard $4e^{-4}$ with an initial random initialization of network weights and we used a batch size of 32 image-caption pairs. The dimension of word embeddings, attention linear layers and decoder RNN have been all set to 512. To avoid network overfitting a dropout at 0.5 was applied. In addition to dropout, the only other regularization strategy we used was early stopping on BLEU score. Since 20th epoch onwards we used "Fine Tuning" of the ResNet based encoder to evaluate possible improvements in the captions generation.

Results are reported in Table 5. In the first row, the results from (Masotti, Croce, and Basili 2018) are reported. In the second row, we report the results of our architecture trained on the same dataset from (Masotti, Croce, and Basili 2018), but considering all available training captions. Then, we evaluated the same architecture on the new dataset. Results confirm the beneficial impact of the new architecture trained over the entire dataset, with a significant improvement especially in term of Cider. Most importantly the beneficial impact of the new available dataset is confirmed by the improved results in terms of BLUE4 and Cider. Figure 3 shows an image which the system associated to the caption "*Un uomo in sella ad una moto su una strada sterrata*" (in English, "*A man riding a motorcycle on a dirt road*"). Moreover, in the same figure, the different areas where the network focused when generating each word are shown.

3.2 Video Captioning

In this evaluation we adopted the model presented in (Laokulrat et al. 2016), which also exploits Attention Mechanisms[6]. This architecture extends the one adopted for Image Captioning used in the previous evaluations. Since a video can be considered as a sequence of images (i.e., the frames), this approach essentially implements a sequence-

6 We used the architecture implemented using PyTorch and available at the following link: `http://github.com/xiadingZ/video-caption.pytorch`

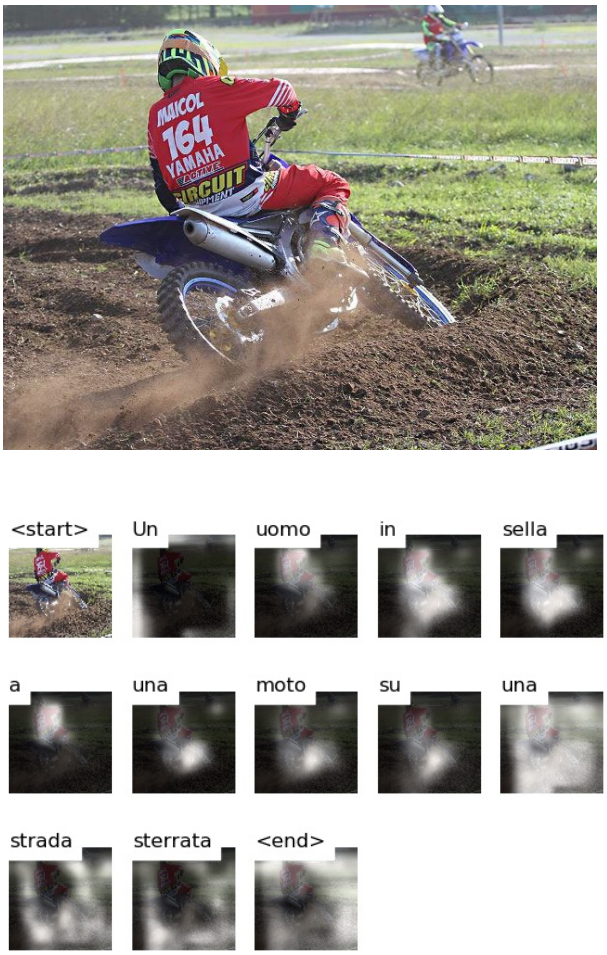


Figure 3
Example of caption generated by the model trained on MSCOCO-it-v2

to-sequence model. In the first encoding stage, a sample of k images are first extracted from the video and encoded by a Convolutional Neural Network (again the ResNet implementation, (He et al. 2016)) to be used in input to a recurrent neural network (again a LSTM network).

Then, in the decoding phase, the LSTM generates word by word the caption by taking into consideration at each time step the hidden state of the network, the hidden state of the previous time step, the word generated at the previous time step and a context vector used to represent a sort of temporal attention (Bahdanau, Cho, and Bengio 2015). This last element allows to weight the contribution of each image in input to focus on those one more important to generate the caption. In the evaluations, the learning rate was set at standard $4e^{-4}$ with an initial random initialization of network weights. The dimension of the word embeddings, the attention linear layers and the decoder RNN has been set at 512 and a recurrent dropout (set to 0.5) is used. The dimension of features encoding the frames is set to 2048 and a a batch size of 128 video-

caption pairs is used. A sample of $k = 30$ was imposed. The network parameters were chosen by selecting those maximizing on average the various measures (BLEU4, CIDER, METEOR and ROUGE-L) on the validation set.

As far as the video captioning task is concerned, we have no reference being the first experimental work done. So we will limit ourselves to make comparisons between the trained network with and without Attention mechanism (Bahdanau, Cho, and Bengio 2015) and in particular focusing on the Attention mechanism. Moreover, we focused on the reliability of the results that can be obtained by the available test material, which is also partially validated.

Table 6
Performances on the Italian Video captioning task.

Name	Test	Bleu_4	Cider	Meteor	Rouge_L
No Attention	v.	0.28	0.34	0.24	0.51
	n.	0.33	0.31	0.25	0.52
With Attention	v.	0.36	0.39	0.26	0.54
	n.	0.35	0.32	0.25	0.54

Figure 4
Example of caption generated by the model trained on MSR-VTT-it. (In English: A man is cooking food in a frying pan)

In table 6 the results of two different systems against two test sets are reported. Results obtained over the validated portion of the test set are denoted by (v.): not validated material is reported in rows (n.). The outcomes confirm the beneficial impact of temporal attention, reported in (Laokulrat et al. 2016): from the first two rows (where the context vector was neglected) to the last two rows, a systematic improvement across

different metrics is reported. An example of captioning obtained by using attention is shown in Figure 3.2. This sounds much interesting as the generalization capability of networks trained on noisy linguistic input is remarkable. Overall, we confirmed the beneficial impact of these resources that, although noisy, trigger the training of large scale networks for Italian, with results comparable with the systems existing for other resource rich languages.

4. Conclusions

In this paper, we proposed two new large scale corpora for Image and Video captioning aimed at enabling the training of effective neural architectures for the Italian language. The work improves the performance on the Image Captioning task for the Italian language and, at the same time, lay the ground-work for future work on Video Captioning in Italian. This last task remains much more difficult than the previous one given the need to capture many more features in the frame sequence than are simply absent over individual images. With our experiments, using models that are not too complex, we hope to support the advancement of the state of the art for the Image and Video caption tasks in Italian. The availability of the two corpora as publicly available resources is expected to trigger more research work on the improvement of the corpus quality as well as on the development of newer neural models through possible language specific architecture.

Acknowledgments

We would like to thank Carlo Gaibisso, Bruno Luigi Martino and Francis Farrelly of the Istituto di Analisi dei Sistemi ed Informatica “Antonio Ruberti" (IASI) for supporting the experimentations through access to dedicated computing resources made available by the Artificial Intelligence & High Performance Computing laboratory.

References

Aafaq, Nayyer, Ajmal Mian, Wei Liu, Syed Zulqarnain Gilani, and Mubarak Shah. 2020. Video description: A survey of methods, datasets, and evaluation metrics. *ACM Computing Surveys*, 52(6):115:1–115:37.

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, (ICLR 2015)*, San Diego, CA, USA, May.

Batra, Vishwash, Yulan He, and George Vogiatzis. 2018. Neural caption generation for news images. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May. European Language Resources Association (ELRA).

Elliott, Desmond, Stella Frank, Khalil Sima’an, and Lucia Specia. 2016. Multi30K: Multilingual English-German image descriptions. In *Proceedings of the 5th Workshop on Vision and Language*, pages 70–74, Berlin, Germany, August. ACL.

Escalante, Hugo Jair, Carlos A. Hernández, Jesús A. González, Aurelio López-López, Manuel Montes-y-Gómez, Eduardo F. Morales, Luis Enrique Sucar, Luis Villaseñor Pineda, and Michael Grubinger. 2010. The segmented and annotated IAPR TC-12 benchmark. *Computer Vision and Image Understanding*, 114(4):419–428.

Feng, Yansong and Mirella Lapata. 2013. Automatic caption generation for news images. *Transactions on Pattern Analysis and Machine Intelligence.*, 35(4):797–812, April.

Funaki, Ruka and Hideki Nakayama. 2015. Image-mediated learning for zero-shot cross-lingual document retrieval. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 585–590, Lisbon, Portugal, September. ACL.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 770–778, Las Vegas, NV, USA, June. IEEE Computer Society.

Hossain, MD. Zakir, Ferdous Sohel, Mohd Fairuz Shiratuddin, and Hamid Laga. 2019. A comprehensive survey of deep learning for image captioning. *ACM Computing Surveys*, 51(6), February.

Laokulrat, Natsuda, Sang Phan, Noriki Nishida, Raphael Shu, Yo Ehara, Naoaki Okazaki, Yusuke Miyao, and Hideki Nakayama. 2016. Generating video description using sequence-to-sequence model with temporal attention. In *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers*, pages 44–52, Osaka, Japan, December.

Lavie, Alon and Abhaya Agarwal. 2007. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, StatMT '07, pages 228–231, Prague, Czech Republic, June. ACL.

Lin, Chin-Yew. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July. Association for Computational Linguistics.

Lin, Tsung-Yi, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, volume 8693 of *Lecture Notes in Computer Science*, pages 740–755. Springer.

Masotti, Caterina, Danilo Croce, and Roberto Basili. 2017. Deep learning for automatic image captioning in poor training conditions. In *Proceedings of the Fourth Italian Conference on Computational Linguistics (CLiC-it 2017)*, Rome, Italy, December.

Masotti, Caterina, Danilo Croce, and Roberto Basili. 2018. Deep learning for automatic image captioning in poor training conditions. *Italian Journal of Computational Linguistics*, 4(1):43–56.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Philadelphia, Pennsylvania, July. ACL.

Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems*, pages 3104–3112, Montreal, Quebec, Canada, December.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, pages 5998–6008, Long Beach, CA, USA, December.

Vedantam, Ramakrishna, C. Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 4566–4575, Boston, MA, USA, June. IEEE Computer Society.

Venugopalan, Subhashini, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond Mooney, and Kate Saenko. 2015. Translating videos to natural language using deep recurrent neural networks. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1494–1504, Denver, Colorado, May–June. ACL.

Vinyals, Oriol, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015*, pages 3156–3164, Boston, MA, USA, June. IEEE Computer Society.

Xu, Jun, Tao Mei, Ting Yao, and Yong Rui. 2016. MSR-VTT: A large video description dataset for bridging video and language. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, pages 5288–5296, Las Vegas, NV, USA, June. IEEE Computer Society.

Xu, Kelvin, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015*, pages 2048–2057, Lille, France, July.

Yao, Li, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher J. Pal, Hugo Larochelle, and Aaron C. Courville. 2015. Describing videos by exploiting temporal structure. In *2015 International Conference on Computer Vision, ICCV 2015*, pages 4507–4515, Santiago, Chile, December. IEEE Computer Society.

PARSEME-It: an Italian corpus annotated with verbal multiword expressions

Johanna Monti*
Università degli Studi di Napoli
L'Orientale

Maria Pia di Buono**
Università degli Studi di Napoli
L'Orientale

The paper describes the PARSEME-It corpus, developed within the PARSEME-It project which aims at the development of methods, tools and resources for multiword expressions (MWE) processing for the Italian language. The project is a spin-off of a larger multilingual project for more than 20 languages from several language families, namely the PARSEME COST Action. The first phase of the project was devoted to verbal multiword expressions (VMWEs). They are a particularly interesting lexical phenomenon because of frequent discontinuity and long-distance dependency. Besides they are very challenging for deep parsing and other Natural Language Processing (NLP) tasks. Notably, MWEs are pervasive in natural languages but are particularly difficult to be handled by NLP tools because of their characteristics and idiomaticity. They pose many challenges to their correct identification and processing: they are a linguistic phenomenon on the edge between lexicon and grammar, their meaning is not simply the addition of the meanings of the single constituents of the MWEs and they are ambiguous since in several cases their reading can be literal or idiomatic. Although several studies have been devoted to this topic, to the best of our knowledge, our study is the first attempt to provide a general framework for the identification of VMWEs in running texts and a comprehensive corpus for the Italian language.

1. Introduction

Multiword expressions (MWEs) represent a difficult lexical construction to identify, model and treat in Natural Language Processing (NLP) tasks, e.g., parsing (Constant, Sigogne, and Watrin 2012), machine translation (Venkatapathy and Joshi 2006; Monti et al. 2013; Mitkov et al. 2018) and keyphrase extraction (Newman et al. 2012), mainly due to their non-compositional property. The lack of compositionality, which concerns the lexical, morphological, syntactic, semantic, pragmatic and statistical level of analysis, namely, (Baldwin 2006), characterizes the behaviour of such linguistic phenomena.

Different types of lexical constructions can be classified as MWEs, with different levels of representation in each language based on their frequency and language-specificity (Salehi, Cook, and Baldwin 2016), e.g., compound nouns are very common in languages such as English (Copestake 2003; Ó Séaghdha 2008) and German (Im Walde, Müller, and Roller 2013), light verb constructions (LVCs) in English (Butt 2010), Persian (Karimi-Doostan 1997), and Italian (Alba-Salas 2002).

* UNIOR NLP Research Group, Dept. of Literary, Linguistic and Comparative Studies - Via Duomo, 219 80138 Napoli, Italy. E-mail: jmonti@unior.it
** UNIOR NLP Research Group, Dept. of Literary, Linguistic and Comparative Studies - Via Duomo, 219 80138 Napoli, Italy. E-mail: mpdibuono@unior.it

Scholars usually do not converge on a unique definition and classification of MWEs nor include in their classifications the same types of MWEs. For our study we refer to the definition of MWEs as "lexical items that: (a) can be decomposed into multiple lexemes; and (b) display lexical, syntactic, semantic, pragmatic and/or statistical idiomaticity[1]" (Baldwin and Kim 2010).

Among these types, verbal multiword expressions (VMWEs) are particularly challenging and, as we will discuss in the next sections, a fine-grained classification is needed. They may present different syntactic structures, e.g., *prendere una decisione* (to make a decision), *decisioni prese precedentemente* (decisions made previously), may be continuous and discontinuous, e.g., *andare e venire* (to come and go) versus *andare in malora* (go to ruin) in *Luigi ha fatto **andare** la società **in malora*** (Luigi made the company go ruin), may have a literal and figurative meaning, e.g., *abboccare all'amo* (to take the bait). Moreover, these units have language-specific features and are generally modelled according to descriptive categories developed by different traditions of linguistic studies.

In this paper, we describe the PARSEME-It VMWE corpus[2], which represents the main outcome of the PARSEME-It project[3], a spin-off project of the European IC1207 COST[4] action PARSEME[5], carried out by the UNIOR NLP Research Group[6]. The main aim of the project is i) to bridge the gap between linguistic precision and computational efficiency in NLP applications by investigating the syntactic and semantic representation of MWEs in language resources and ii) the integration of MWE analysis in syntactic parsing and translation technology. Deliverables include mainly enhanced monolingual language resources (lexicons, grammars and annotated corpora) in Italian or multilingual linguistic resources with the Italian language. The UNIOR NLP Research group, together with the language leaders working on other languages, has contributed to developing the general and language-specific guidelines for the PARSEME annotation process.

We discuss related researches in linguistic studies on VMWEs and more in general in MWE processing, including a description of the PARSEME COST Action and its aims (Section 2). Then, the PARSEME-It corpus (Section 3) is introduced. In Section 4, we present the VMWE categories included in the annotation scheme and in Section 5 the annotation guidelines, the identification tests and decision trees used. The description of the annotation process (Section 6) and annotation issues (Section 7), the analysis of productive categories and borderline cases (Section 8) follow. Finally, we discuss conclusions and future work (Section 9).

2. Related Work

As a diverse and complex phenomenon present in all natural languages (Jackendoff 1997; Sag et al. 2002), MWEs have attracted the interest of many disciplines.

1 As defined by Lyse and Andersen (2012), "statistical idiomacity is the phenomenon of particular combinations of words occurring with markedly higher frequency in comparison to alternative phrasings of the same concept".
2 https://github.com/UNIORNLP/PARSEME-It-Corpus
3 https://sites.google.com/view/parseme-it/home
4 https://www.cost.eu
5 https://typo.uni-konstanz.de/parseme
6 https://sites.google.com/view/unior-nlp-research-group/home?authuser=0

Recently Constant et al. (2017) proposed a classification including MWE categories which are non-exhaustive and may overlap:

- **Idiom**: a group of lexemes whose meaning is established by convention and cannot be deduced from the individual lexemes composing the expression (e.g., *tirare le cuoia* → to kick the bucket).
- **Light-verb construction**: it is formed by a head verb with light semantics that becomes fully specified when combined with a (directly or indirectly) dependent predicative noun (e.g., *fare una passeggiata* → to take a walk).
- **Verb-particle construction**: it comprises a verb and a particle, usually a preposition or adverb, which modifies the meaning of the verb and which needs not be immediately adjacent to it (e.g., *buttare giù* → to swallow). Verb-particle constructions are also referred to as *phrasal verbs*.
- **Compound**: a lexeme formed by the juxtaposition of adjacent lexemes, occasionally with morphological adjustments (e.g., *carta di credito* → credit card). Compounds can be subdivided according to their syntactic function. Thus, nominal compounds are headed by a noun (e.g., *lettera aperta* → open letter) whereas noun compounds and verb compounds are concatenations of nouns (e.g., *treno merci* → freight train) or verbs (e.g., *lasciar andare* → let go). Some authors (Stymne, Cancedda, and Ahrenberg 2013; Shapiro 2016; Gagné and Spalding 2009) refer to closed compounds when they are composed of a single token (e.g., *banconota* → banknote), and open compounds when they consist of lexemes separated by spaces or hyphens (e.g., *fuggi-fuggi* → rush).
- **Complex function word**: it is a function word formed by more than one lexeme, encompassing multiword conjunctions (e.g., *non appena* → as soon as), prepositions (e.g., *fino a* → up until), and adverbials (e.g., *in linea di massima* → by and large).
- **Multiword named entity**: a multiword linguistic expression that rigidly designates an entity in the world, typically including people, organizations, and locations (e.g., *Organizzazione delle Nazioni Unite* → United Nations).
- **Multiword term**: a multiword designation of a general concept in a specific subject field (e.g., *missione scientifica a breve termine* → short-term scientific mission).

More specifically, MWEs are characterized by a set of properties, pointed out by Markantonatou et al. (2018), which increase the difficulty of their automatic processing:

1. **Semantic non-compositionality**. In numerous cases, the meaning of VMWEs cannot be deduced on the basis of their syntactic structure and of the meanings of their components. For instance, the meaning of *me lo ha detto l'uccellino* (a bird told me that) as *qualcuno me lo ha detto in segreto* (someone told me that in secret) cannot be deduced by the meanings of *dire* (tell) and *uccellino* (little bird).

2. **Lexical and grammatical inflexibility**[7]**.** Lexical and syntactic constraints of VMWEs may be unpredictable, e.g., *ha messo il carro davanti ai buoi* (lit. 'he put the cart in front of the oxen' → he put the cart in front of the horse) e non **ha messo i carri davanti ai buoi* (lit. 'he put the carts in front of the oxen') or **ha messo il calesse davanti ai buoi* (lit. 'he put the calesh in front of the oxen').
3. **Regular variability.** Even though VMWEs present lexical and grammatical inflexibility, they may present some regular variability as well, e.g., *prendere una decisione* (to make a decision): *La decisione che prendemmo* (the decision we made).
4. **Discontinuity.** Elements in a VMWE may not be adjacent, e.g., *fornire un contributo* (to make a contribution): *Ha fornito un rilevante contributo al progetto* (He made a significant contribution to the project).
5. **Categorical ambiguity.** VMWEs sharing the same syntactic structure and lexical choices, as in *fare un discorso* (to give a speech) and *fare un dolce* (to make a cake), may belong to different categories, i.e., *fare un discorso* is a light verb construction, while *fare un dolce* is not an MWE in that the element co-occurring with the verb is a concrete noun (Ninio 2011).
6. **Syntactic ambiguity.** VMWE occurrences may be syntactically ambiguous, e.g., *giù* is an adverb in *buttare giù la palla* and a particle in *buttare giù un boccone*, where it takes the meaning of *to swallow*.
7. **Literal-idiomatic ambiguity.** Some VMWEs may present both a literal and idiomatic meaning, e.g., *Ha preso il toro per le corna* (lit. 'he took the bull by its horns' → grasp the nettle).
8. **Non-literal translatability.** VMWEs usually may not be translated by means of a word-for-word process. *Il mattino ha l'oro in bocca* (lit. 'the morning has gold in its mouth' → the early bird catches the worm).
9. **Cross-lingual divergence.** VMWE behaviours change across different languages, as they are the result of different linguistic traditions. For instance, in Germanic languages *off* has a status of stand-alone word and forms verb-particle constructions, while in Slavic languages is a prefix and becomes an inherent part of verbal lexemes (Markantonatou et al. 2018) as in (PL) *wyłączyć* 'part. connect' → turn off).
10. **Wordplay proneness.** VWMEs allow playful usage and creativity in some specific contexts. For instance, *vuole che rimetta tutto nel sacco dopo che l'ho svuotato* (lit. 'He wants me to put everything again in the bag after I have emptied it') from *svuotare il sacco* with the idiomatic meaning of *to blow the whistle*.

Two threads of research are relevant to our work: (i) linguistic studies on Italian VMWEs, mainly with the contribution of scholars working on the Italian language; and (ii) MWE Processing. The former aims at presenting current research outputs in

7 Sheinfux et al. (2019) provide an interesting discussion on the concept of inflexibility of VMWEs, starting from the work by Gibbs et al. (1989) and Nunberg et al. (1994).

contrastive/comparative analyses and synchronic and diachronic studies. The latter takes into account computational researches on MWE processing, as the developed corpus is intended to improve the automatic processing of these linguistic phenomena.

Linguistic Studies on VMWEs. Several scholars have investigated different categories of Italian VMWEs, focusing on both syntactic and semantic aspects. Among these works, we may distinguish contrastive and comparative analyses, and synchronic and diachronic studies.
In the first group, most of the scholars propose a comparison with Germanic languages (Mateu and Rigau 2010), mainly for describing verb-particle constructions, that represent a very common phenomenon in this family.
On the other hand, synchronic and diachronic studies include analyses of: (i) verb-particle constructions (Simone 1997; Masini 2005; Iacobini and Masini 2005; Quaglia and Trotzke 2017), (ii) idiomatic constructions (Tabossi, Arduino, and Fanari 2011; Vietri 2014c) with either ordinary or support verbs (Vietri 2014a), (iii) support, or light, verbs, which represent a wider phenomenon and, for this reason, they have been largely analysed (La Fauci 1980; D'Agostino and Elia 1998; Cicalese 1999; Alba-Salas 2004; Jezek 2004; Quochi 2007; Cicalese et al. 2016).
Reflexive verbs in Italian have been investigated as occurrences of non-local anaphora (Reuland 1990) and considering their syntactic classification (Carstea-Romascanu 1977).
To the best of our knowledge only a limited number of monolingual language resources with multiwords for the Italian language have been developed such as a dictionary for Italian idioms (Vietri 2014b), a series of example corpora and a database of MWEs represented around morphosyntactic patterns (Zaninello and Nissim 2010), or a corpus annotated with Italian MWEs of a particular class: verb-noun expressions such as *fare riferimento*, *dare luogo* and *prendere atto* (Taslimipoor et al. 2016). With reference to Italian word combinations, it is worth mentioning the CombiNET project[8], which represents an important contribution to MWE extraction from Italian corpora (Nissim, Castagnoli, and Masini 2014), and SYMPAThy, a new approach to the extraction of this type of occurrences (Lenci et al. 2014). At the time of writing, therefore, the PARSEME-It VMWE corpus represents the first sample of a corpus, which includes several types of VMWEs, specifically developed for NLP applications.

MWE Processing. MWEs have been the focus of the PARSEME COST Action, which enabled the organization of an international and highly multilingual research community (2015). This community launched in 2017 the first edition of the PARSEME shared task on automatic identification of VMWEs (Savary et al. 2017), which was replicated in 2018 (Ramisch et al. 2018) with the aim of developing universal terminologies, guidelines and methodologies for several languages, including the Italian language. To increase the computational efficiency of Natural Language Processing (NLP) applications, PARSEME focused on a special class of Multiword Expressions, namely VMWEs. The main outcomes include unified definitions and annotation guidelines for several types of VMWEs, as well as a large multilingual openly available VMWE annotated corpus.

In the first edition, eighteen languages were addressed, including 4 non-Indo-European languages. The task was co-located with the 13th Workshop on Multiword Expressions (MWE 2017) (Markantonatou et al. 2017), which took place during the

8 https://sites.google.com/site/enwcin/home

European Chapter of the Association for Computational Linguistics (EACL 2017). A corpus of 5.5 million tokens and 60,000 VMWE annotations in the 18 languages was released and distributed under different versions of the Creative Commons license.

In the second edition the annotation methodology was enhanced and the set of languages was changed reaching twenty languages. The task was co-located with the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018) (Savary et al. 2018) at COLING 2018 (Santa Fe, USA). A corpus of 6 million tokens and 79,000 VMWE annotations in the 20 languages was released and also, in this case, it is distributed under different versions of the Creative Commons license.

A focused overview of how MWEs are handled in NLP applications, with particular attention to the nature of interactions between MWE processing and downstream applications in NLP, such as MWE parsing and Machine Translation (MT) can be found in Constant et al. (2017).

With reference to MT, MWE-aware technologies have been proved successful in several cases (Pal, Naskar, and Bandyopadhyay 2013; Cap et al. 2015). In order to improve the quality of translation, various strategies, depending on the MT paradigm, have been proposed to overcome problems related to MWE processing (Ren et al. 2009; Kordoni and Simova 2014; Ramisch, Besacier, and Kobzar 2013; Barreiro et al. 2014). Also in neural approaches to MT, some recent contributions show that the proper handling of MWE improves the translation of MWEs by adding synthetic MWE data to the training corpora (Rikters and Bojar 2017) or by annotation and data augmentation, using external linguistic resources (Zaninello and Birch 2020).

Finally, the workshop series titled Multiword Units in Machine Translation and Translation Technology (MUMTTT) (Monti et al. 2013; Pastor et al. 2015; Monti et al. 2018; Pastor et al. 2019) and the recent volume on the same topic (Mitkov et al. 2018) provide an overview of state-of-the-art research in this field and highlight the importance of proper computational treatment of these lexical units in MT and translation technology (TT).

Besides NLP tasks, cross-lingual studies of multiwords and automatic extraction of translation equivalents represent an important field of research. With the aim of building MWE repositories, Wehrli and Villavicencio (2015) propose an extraction methodology based on aligned corpora for English, Portuguese and French. They combine a symbolic parser with a high-recall statistically-based extraction method and identify correspondences in the language pairs using alignment and distributional methods (de Caseli et al. 2010; Laranjeira et al. 2014).

Acknowledging the diversity of idiomatic structures,Villavicenzio et al. (2004) propose a framework for the cross-lingual collection of idioms and mapping of their equivalent parts which allows the identification of similarity at semantic, syntactic and lexical levels.

Statistical methods have been applied to parallel corpora (Wehrli and Villavicencio 2015) to evaluate their cross-lingual applicability for idiomatic pattern identification, while Taslimipoor et al. (2016) improve the performance of monolingual association measures by augmenting them with information about translation equivalents and using them to produce a ranking of expressions according to their idiomaticity.

3. PARSEME-It VMWE Corpus

This section outlines the **PARSEME-It VMWE corpus** (version 1.1), annotated with VMWEs for the Italian language. As described in the previous sections, the corpus is

the main outcome of the PARSEME-It project together with the general and language-specific guidelines for the PARSEME annotation process.

The corpus is based on a selection of texts taken from the *PAISÀ* corpus of Italian web texts[9] (Lyding et al. 2014). We chose this corpus because its documents are:

1. representative of different web sources, e.g., Wikibooks, Wikinews, Wikiversity, and several blog services from different websites, collected in 2010 by means of a Creative Commons-focused web crawling, and a targeted collection of documents from specific websites;
2. dedicated to no specific technical domain, free from copyright issues, so as to be compatible with an open license;
3. annotated in CoNLL format, i.e. lemmatized, POS-tagged and annotated with syntactic dependencies.

For our annotation task, we selected a sub-corpus formed by 15,728 sentences (corresponding to 430,789 tokens) randomly taken from blogs, Wikipedia and Wikinews. Due to the heterogeneous sources, e.g., social media, blogs, forum posts, consumer reviews, texts present variable characteristics: inconsistent punctuation and capitalization, use of slang and technical jargons, specific syntactic constructions related to genres. Nevertheless, the corpus was kept in its original state and therefore no errors or inconsistencies were corrected. The automatically pre-annotated information in the original corpus, namely morpho-syntactic and dependency annotations[10], were kept to ease the annotation work regarding the identification of VMWEs, but we asked annotators not to overestimate the system's performances, and to review the whole text, not only the pre-annotated candidates, namely all the verbs (V). A dedicated tag in FLAT, the web-based annotation environment used in the project (Section 6), was defined for this purpose.

The objective was to have a final corpus of at least 3,500 annotated VMWEs. Since the density of VMWEs in the corpus is highly dependent on the particular language, as well as text choice and genre, we were not able to make any reliable estimation of the corpus size needed to reach this goal from the beginning of the task.

4. VMWE Categories

For the Italian VMWE annotation task, according to the PARSEME guidelines, multiword expressions are understood as (continuous or discontinuous) sequences of words with the following compulsory properties:

- their component words include a head word and at least one other syntactically related word. Most often the relation they maintain is a syntactic (direct or indirect) dependency but, for instance, it can also be a coordination.

9 https://www.corpusitaliano.it/en/

10 The tag sets for such annotation have been developed by the Istituto di Linguistica Computazionale "Antonio Zampolli" (ILC-CNR) and the University of Pisa in the framework of the TANL (Text Analytics and Natural Language processing) project.

- they show some degree of orthographic, morphological, syntactic or semantic idiosyncrasy with respect to what is considered general grammar rules of a language.
- at least two components of such a word sequence have to be lexicalized.

Only the lexicalized components[11] were annotated and open slots[12] ignored, as in *prendere qualcuno di sorpresa*, only *prendere ... di sorpresa* are annotated, while *qualcuno* is not because it can be replaced by a noun or a pronoun. Collocations, i.e., word co-occurrences whose idiosyncrasy is of statistical nature only (e.g., *the graphic shows, drastically drop*, etc.), were excluded from annotation as well. Therefore, the VMWE which have been annotated for the Italian language are:

1. **Light verb constructions** (LVC), which typically consist of a verb and a noun or prepositional phrase, e.g., *fare una domanda* (to make a question). The verb has a purely syntactic operator function (performing an activity or being in a state), whereas the noun is predicative, often referring to an event (e.g., decision, visit) or a state (e.g., fear, courage). This category has two subclasses: i) LVCs in which the verb is semantically totally bleached (**LVC.full**), e.g. *fare una passeggiata* (to have a walk) and ii) LVCs in which the verb adds a causative meaning to the noun (**LVC.cause**), e.g. *dare il mal di testa* (to give a headache);
2. **Idioms** (VID), which have at least two lexicalized components including a head verb and at least one of its arguments, e.g., *tirare le cuoia* (to kick the bucket), *piovere a catinelle* (to rain cats and dogs);
3. **Inherently reflexive verbs** (IRV), account for those reflexive verbal constructions (a) which are never used without a reflexive clitic pronoun e.g., *suicidarsi* (to suicide), or (b) when the IRV and non-reflexive versions have clearly different senses or subcategorization frames e.g., *farsi* (to take drugs) while the non-pronominal form, *fare*, means *to make*.
4. **Verb particle combinations** (VPC), which are formed by a lexicalized head verb and a lexicalized particle dependent on the verb. The meaning of the VPC is non-compositional. Notably, the change in the meaning of the verb goes significantly beyond adding the meaning of the particle, e.g., *fare fuori* (lit. 'to do out' $\rightarrow$ to kill). This type of construction is very frequent in English, German, Swedish, Hungarian, but we can find it also in Italian. The VPC category is split in two subcategories as well: fully non-compositional VPCs (**VPC.full**), in which the particle totally changes the meaning of the verb as in *fare fuori* and semi non-compositional VPCs (**VPC.semi**), in which the particle adds a partly predictable but non-spatial

11 According to Savary and Cordeiro (2018), the lexicalized components of an MWE are those which are always realized by the same lexeme. For instance in *to pay a visit* the head verb is always a form of *pay* and the object is always *visit*: these two elements are therefore lexicalized components of the VMWE.

12 An open slot (Savary and Cordeiro 2018) is a component of a compulsory argument which can be realized by a free lexeme taken from a relatively large semantic class. If we consider again the example of the VMWE *to pay a visit*, an open slot is represented by the determiner *a*, which can be freely replaced, as in *paid many visits*.

meaning to the verb like in *tirare avanti* (to go on) since the preposition *avanti* no longer owns its spatial meaning (forward).

5. **Multi Verb Constructions** (MVC), which are composed by a sequence of two adjacent verbs (in a language-dependent order), a governing verb (also called a vector verb) and a dependent verb (also called a pole/polar verb), e.g. *lasciar perdere* (lit. 'let lose' → to forget).

6. **Inherently Adpositional Verbs** (IAV), which consist of a verb or VMWE and an idiomatic selected preposition or postposition that is either always required like in *appartenere **a*** (to belong to) or, if absent, changes the meaning of the verb of VMWE significantly, like in *contare **su*** where the preposition *su* is required to express the meaning of 'to rely on' compared to the verb without the preposition which means *to count*. It is a special optional and experimental category, corresponding to what is sometimes called in English *prepositional verbs*[13].

Besides these categories, shared by all languages involved in the PARSEME COST Action, language specific categories have been introduced in edition 1.1 of the PARSEME Shared Task. For the annotation of the Italian language, the Inherently Clitic Verbs (LS.ICV) category was proposed and carefully defined by means of linguistic tests that allow to distinguish this category from IRVs.

A language specific category: Inherently clitic verbs. LS.ICVs are an extremely rich and varied VMWE category for some Romance languages, and they are particularly frequent in the Italian language (Masini 2015).

LS.ICVs together with IRVs are pronominal verbs (De Mauro 2000): they are formed by a full verb combined with one or more non-reflexive clitics that represent the pronominalization of one or more complements (CLI) (Viviani 2006; Berruto 1987). LS.ICV is annotated when (a) the verb never occurs without one non-reflexive clitic, e.g. *entrarci* colloquial form, or (b) when the LS.ICV and the non-clitic versions have clearly different senses or subcategorization frames, like *entrar**ci*** when it means *to be relevant to something*, while the intransitive form of the verb *entrare* means *to enter*.
It is often challenging to distinguish LS.ICV from IRV, particularly because some clitics may be ambiguous, like *se/si* (Cinque 1988; Cordin 2001; Pescarini 2015) which is a polyfunctional clitic pronoun and grammatical marker (and can have a reflexive, reciprocal, impersonal, passivizing, aspectual, and middle function).

The following verbs are annotated as LS.ICV:

- The verb without the CLI does not exist, e.g., *infischiarsene* (do not worry about) vs **infischiare, *infischiarsi*;
- The verb without the CLI does exist, but has a very different meaning as in *prenderle* (lit. 'to take them' → to be beaten) vs *prendere* (to take) or *prenderci* (lit. 'to take it' → to grasp the truth) vs *prendere* (to take);

13 Schneider, N., Green, M., 2015, New Guidelines for Annotating Prepositional Verbs, https://github.com/nschneid/nanni/wiki/Prepositional-Verb-Annotation-Guidelines

- The verb has more than one CLI of which the second one is an invariable object complement, like in *fregarsene* (lit. 'to matter self of-it' → do not care about) or *infischiarsene* (do not worry about);
- The verb has two non-reflexive invariable CLIs, like in *farcela* (lit 'to make there it'→ to succeed);
- The verb has a different meaning with respect to an intensive use of the same two non-reflexive invariable CLIs, like in *andarsene* (lit. 'to go away self from-there' → to die) vs *andarsene* (to go away) or *bersela* (lit. 'to drink self it', → to believe) vs *bersela* (to drink it).

A language-specific decision tree to annotate LS.ICV occurrences was developed, as described in Section 5.

5. Annotation Guidelines and Decision Trees

The PARSEME annotation guidelines have been developed with the aim of delivering general definitions and prescriptions for the annotation of VMWEs in all languages involved in the shared task, but, at the same time, of allowing language-specific descriptions of these linguistic phenomena (Savary et al. 2017). We describe here the guidelines and methodologies used for the second annotation trial of the Shared Task, which introduced some novelties to cover a wider range of VMWEs, left apart in the first edition. The improvements of the second edition were particularly valuable for the data collection carried out on the Italian language, because they addressed some peculiarities of the Italian language which were not considered previously, such as the LS.ICV category.

For the second edition of PARSEME annotation task, the following categories were identified:

1. two **universal** categories, common to all languages involved in the task and hold both LVC categories, namely **LVC.full**, and **LVC.cause**, and idioms (**VID**);
2. three **quasi-universal** categories, relevant for some languages or language families but non-existent or very exceptional in others. This category encompasses **IRV**, the two subclasses of VPCs, namely **VPC.full** and **VPC.semi** and finally **MVC**;
3. the **optional** VMWEs category **IAV**;
4. **language-specific** categories, defined for a particular language in separate documentation, as in the case of the Italian language, the **LS.ICV**.

5.1 Identification tests

In order to ease the identification and categorisation task of VMWEs, a decision method was devised with generic and language-specific tests. Generic tests consider general criteria that are valid for all languages, while language-specific tests consider structural, lexical, morphological and syntactic features that are specific for the individual languages. Each iteration of the annotation process includes three steps:

1. Identification of a VMWE candidate, i.e., a combination of a verb with at least one other word, which is a potential VMWE;
2. Identification of the lexicalized elements of the expression;
3. Assignment of the VMWE to one of the VMWE categories, using general and language specific tests.

The first two steps largely rely on the annotator's linguistic intuition and knowledge. As reported by Markantonatou et al. (2018), the identification of a VMWE, regardless of the category, may be accomplished by five generic tests on compositional aspects.

- Test 1 [CRAN] : Presence of a cranberry word, e.g., *mangiare a ufo* (to eat without paying) → *a ufo* is not a stand-alone word;
- Test 2 [LEX] : Lexical inflexibility, e.g., *non dire gatto se non ce l'hai nel sacco* (lit. 'don't say cat if you don't have in the sack' → don't count on something before it happens) vs **non dire cane se non ce l'hai nel sacco* (lit. 'don't say dog if you don't have in the sack');
- Test 3 [MORPH] : Morphological inflexibility, e.g., *andare a letto con le galline* (lit. 'to go to bed with the hens' → to go to bed early) vs **andare a letto con la gallina* (to go to bed with the hen);
- Test 4 [MORPHOSYNT] : Morpho-syntactic inflexibility, e.g., *farò del mio meglio* (I will do my best) vs **Farò del tuo meglio* (*I will do your best);
- Test 5 [SYNT] : Syntactic inflexibility, e.g., *vivi e lascia vivere* (live and let live) → **lascia vivere e vivi* (let live and live).

Besides these five tests, a specific hypothesis has been formulated to identify LVC candidates, which do not pass Tests 1 and 3-5 and for which Test 2 is hard to apply due to their high productivity, even though they present some restrictions.
LVC hypothesis: In a verb+(prep)+noun candidate the verb is a pure syntactic operator and the noun expresses an activity or a state, e.g. *fare un discorso* (to make a speech). If a candidate group passes any of the previous tests, it can be annotated as VMWE. To confirm the LVC hypothesis a specific test, namely Test 6 described in Section 5.2, has to be applied.

5.2 Category Decision Trees

In order to select a category for the identified VMWEs, a decision tree formed of both structural and category tests is provided (Figure 1). The decision tree is formed by a set of tests which help the annotator to identify and annotate VMWE candidates.

Tests S.1-S.4 (prev. 6-8) are structural, which means that the categorization is based on the syntactic structure of VMWE canonical form and defined by means of four tests:

- Test S.1 (prev. 6) [1HEAD]: Presence of a unique verb functioning as the syntactic head of the whole expression, e.g., *fare fuori* (lit. 'to make out' → to kill) → *fare* is the head and *fuori* is a particle depending on it;
- Test S.2 (prev. 7) [1DEP]: Among the phrases dependent on the head verb exactly one contains lexicalised components, e.g., *prendere in considerazione*

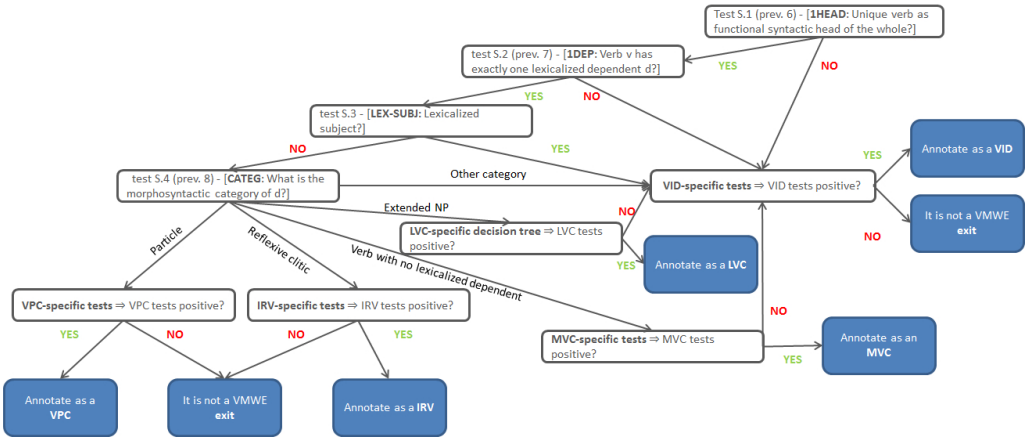


Figure 1
Decision tree for VMWE categorization

(to take into consideration) → the single dependent is a prepositional phrase, *in considerazione*;

- Test S.3 [LEX-SUBJ]: a single lexicalized (functional) syntactic dependent of the head verb is its subject, e.g., *me lo ha detto l'uccellino* (a bird told me) → *l'uccellino* (a bird) is the subject of *ha detto*;
- Test S.4 (prev. 8) [CATEG] Morphosyntactic category of the verb's dependent. This is a closed list of different values, namely (i) reflexive clitic (refl), e.g., *suicidarsi* (to suicide), (ii) particle (part), e.g., *far fuori* (lit. 'to make out' → to kill), (iii) no lexicalized dependent, e.g., *lasciar andare* (lit. 'to let go' → to unhand), (iv) adposition (preposition or postposition, as opposed to a particle), e.g., *confidare su* (to trust in), (v) extended nominal phrase, e.g., *rompere il silenzio* (to break the silence) → *il silenzio* is a noun phrase composed of an article and a singular noun, (vi), e.g., adjective *vedere nero* (to see black), (vii) adverb, e.g., *fare passi avanti* (lit. 'to make steps forward' → to progress), (viii) pronoun, e.g., *farcela* (lit. 'to make it' → to manage), (ix) verb with a lexicalised dependent including fully lexicalized clauses, e.g., *non avere peli sulla lingua* (lit. 'not have hair on the tongue' → to be outspoken), (x) other.

The other tests, i.e., VID-specific tests, LVC-specific decision trees, IRV-specific tests, VPC-specific tests, and MVC-specific tests are categorial and allow to categorize each of the classes identified initially. A complete analysis of those decision trees is provided by Markantonatou et al. (2018). Among these tests, we present the one created to classify the Italian language-specific category of ICVs.

The annotation of LS.ICV was performed following a specific decision tree[14] (Figure 2).

Three types of LS.ICV have been identified:

14 http://parsemefr.lif.univ-mrs.fr/parseme-st-guidelines/1.1/?page=060_Language-specific_tests/015_Inherently_clitic_verbs__LB_LS.ICV_RB_

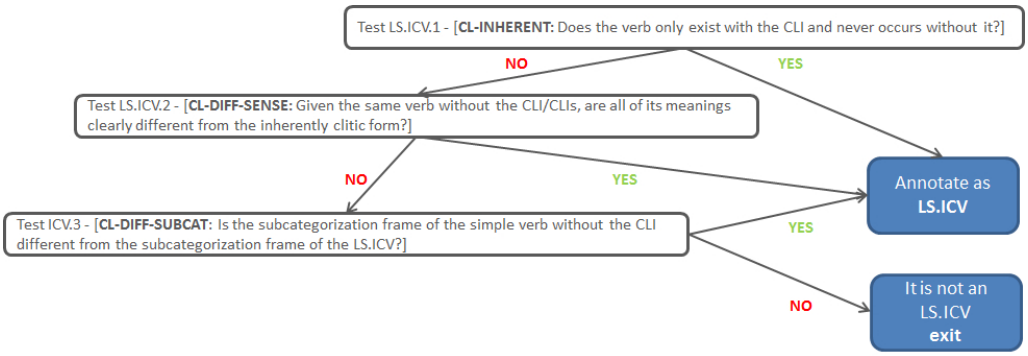


Figure 2
LS.ICV-specific decision tree

- Test LS.ICV.1 [CL-INHERENT]: the verb does exist only in the form with the clitic, e.g., *infischiarsene* (not worry about) vs *infischiare.
- Test LS.ICV.2 [CL-DIFF-SENSE]: the verb without the clitic exists but has a different meaning, e.g. *prenderle* (lit. 'to take them' → be beaten) vs *prendere* (take).
- Test LS.ICV.3 [CL-DIFF-SUBCAT]: the subcategorization frame[15] of the verb without the clitic is different from the subcategorization of the same verb with the clitic, e.g., *X se la prende con Y* (X is angry with Y) vs *X prende Y* (X takes Y).

In the training corpus 20 different LS.ICV were annotated manually, such as *farcela*, *rimetterci*, *fregarsene* among others.

6. Annotation Process and Inter-Annotator Agreement

For the annotation of the PARSEME-It VMWE corpus we used FLAT[16], a web-based linguistic annotation environment based around the FoLiA format[17] a rich XML-based format for linguistic annotation. FLAT is a document-centric tool that fully preserves and visualises document structure and allows users to view annotated FoLiA documents and enrich these documents with new annotations (Figure 3)[18]: it offers a wide variety of linguistic annotation types supported through the FoLiA paradigm.

The annotation task for the Italian language was performed in five different stages:

15 A subcategorization frame of a verb describes how syntactic arguments are realized as the verb's dependents, for a given sense of the verb. A subcategorization frame indicates morphological and syntactic features of a verb's dependents, namely the required prepositions, postpositions and case markers of the subject, direct and oblique objects.
https://parsemefr.lis-lab.fr/parseme-st-guidelines/1.1/?page=glossary#subcat-frame

16 FLAT is an open-source software developed at the Centre of Language and Speech Technology, Radboud University Nijmegen and is licensed under the GNU Public License v3 -
http://flat.readthedocs.io/en/latest/

17 http://proycon.github.io/folia

18 Translation of the example in fig. 3: *Perhaps, inadvertently, Monckton and Fielding did not make such a foolish request*.

1. The PARSEME Annotation guidelines were agreed on[19] and examples for the Italian language were added in order to ease the annotation task by the Italian annotators. To this end, a two-phase pilot annotation in Italian was

Figure 3
Example of annotated data in FLAT

carried out. This step was useful in identifying the Italian VMWE categories to be annotated, but also to promote cross-language convergences with the other languages foreseen in the shared task. Each pilot annotation phase provided feedback from annotators and was followed by enhancements of the guidelines, corpus format and processing tools.

2. A pre-processing step of the PAISÀ corpus was needed. Although the tokenization follows the original tokenization of the PAISÀ corpus, some pre-processing has been applied to the original files of the corpus in order to split compound prepositions (*dei, nei, delle*, etc.), e.g., *dei* is split in the preposition *di* + the determiner *i* to allow the annotation of the preposition only, for instance, as lexicalised component of IAVs. To this end, we added new tokens corresponding to the components of the compound prepositions (see example below[20] in Table 1-2) and we also realigned all the dependency index: the heuristic being used is that the preposition is the head of the prepositional article (all tokens pointing to the prepositional article will point to the preposition in the split version and the determiner also points to the preposition). For instance the original CONLL-U sentence in Table 1[21]. In addition, we also introduced the

Table 1
Original CoNLL-U sentence

Rank	Surf	Lemma	PosG	PosF	Morph	DepIndex	DepLabel
1	Perchè	Perchè	C	CS	_	4	mod
2	la	il	R	RD	num=s|gen=f	3	det
3	ragione	ragione	S	S	num=s|gen=f	4	subj
4	sta	stare	V	V	num=s|per=3|mod=i|ten=p	0	ROOT
5	nel	in	E	EA	num=s|gen=m	4	comp
6	mezzo	mezzo	S	S	num=s|gen=m	5	prep
7	no	no	B	BN	_	4	neg
8	?	?	F	FS	_	4	punc

`SpaceAfter=No` tag on the word preceeding a clitic belonging to the

19 http://parsemefr.lif.univ-mrs.fr/parseme-st-guidelines/1.0/?page=home
20 source: https://gitlab.com/parseme/sharedtask-data//tree/master/1.1/IT
21 For more information about CoNLL-U format, see https://universaldependencies.org/format.html

Table 2
Transformed sentence

Rank	Surf	Lemma	PosG	PosF	Morph	DepIndex	DepLabel
1	Perché	Perché	C	CS	_	4	mod
2	la	il	R	RD	num=s|gen=f	3	det
3	ragione	ragione	S	S	num=s|gen=f	4	subj
4	sta	stare	V	V	num=s|per=3|mod=i|ten=p	0	ROOT
5-6	nel	_	_	_	_	_	_
5	in	in	E	E	_	4	comp
5	il	il	R	RD	_	5	det
7	mezzo	mezzo	S	S	num=s|gen=m	5	prep
8	no	no	B	BN	_	4	neg
9	?	?	F	FS	_	4	punc

same token, e.g., *lavar-si*. These are annotated as two separate words in the original corpus.

3. The annotation task of the training set (approx. 14,000 sentences) was manually performed in running texts using the FLAT environment by five Italian native speakers with linguistic background. Each annotator was given a certain number of files, containing 1,000 sentences in CoNLL format. All the doubts about the annotation were collected in a shared file and discussed during the annotation phase. Difficulties in annotating VMWE mainly concerned (i) the boundaries of the VMWE such as in *Sei ovviamente nel pieno diritto di esprimere [...]* where it is difficult to decide if the VMWE should be *sei ... nel ... diritto* or *sei ... nel pieno diritto*, (ii) the category attribution concerning, for instance, the *fare* + *N* VMWE type since in some cases the category is LVC such as in *fare rumore* and in some others is VID such as in *fare schifo*, (iii) the identification of nested VMWEs like in *Mi guardo bene* where the annotator has to decide if in the VID *guardarsi bene* there is also a IRV *guardarsi* or not.
4. A few files were double-annotated to evaluate the inter-annotator agreement (IAA).
5. Further 1,000 sentences were used as test-set during the shared task. The VMWE annotations were automatically annotated by the systems that took part in the shared task and performed according to the same guidelines.

The current version of the PARSEME-IT corpus (1.1) represents a substantial improvement (Monti et al. 2018) in comparison to its first version (Monti, di Buono, and Sangati 2017) both in terms of categories of VMWEs taken into account for the annotation and total amount of annotated VMWEs.

Table 3 presents the statistics of the various categories of VMWEs in the PARSEME-It corpus 1.0[22], where only five categories were taken into account, namely ID (corresponding to the current VID category), IReflV (corresponding to the current IRV category), LVC, VPC and a OTH category for the VMWEs which could not be included in the previous categories. This version of the PARSEME-It corpus encompasses 1,954 VMWE annotations.

Table 3
PARSEME-It corpus version 1.0

Sent.	Tokens	VMWE	ID	IReflV	LVC	VPC	OTH
15728	387325	1954	913	580	395	62	4

Table 4, instead, shows information about the corpus version 1.1 released for the second edition of the PARSEME shared task, where a total amount of 3,754 VMWEs are annotated.

Table 4
PARSEME-It corpus version 1.1

Lang-split	Sent.	Tokens	Avg. length	VMWE	VID	IRV	LVC	VPC	IAV	MVC	LS.ICV
IT-train	13555	360883	26.6	3254	1098	942	691	66	414	23	20
IT-dev	917	32613	35.5	500	197	106	119	19	44	6	9
IT-test	1256	37293	29.6	503	201	96	129	23	41	5	8
IT-Total	15728	430789	27.3	4257	1496	1144	939	108	499	34	37

PARSEME-It VMWE corpus 1.1. includes i) the manually annotated training set, ii) manually annotated development set and finally iii) the automatically annotated test set. For each of those morphosyntactic data (parts of speech, lemmas, morphological features and/or syntactic dependencies) are also provided.

The data have been annotated using the official parseme-tsv format (Figure 4), adapted from the CoNLL format.

In the official parseme-tsv format, as described in Savary et al. (2017), the information about each token is represented by 4 tab-separated columns featuring:

- the position of the token in the sentence or a range of positions (e.g., 1-2) in case of multiword tokens such as contractions;
- the token surface form;
- an optional flag indicating that the current token is adjacent to the next one;
- an optional VMWE code composed of the VMWE's consecutive number in the sentence and – for the initial token in a VMWE – its category (e.g., 2:ID

22 The corpus is provided in the parseme tsv format, inspired by the CONLL-U format https://typo.uni-konstanz.de/parseme/index.php/2-general/184-parseme-shared-task-format-of-the-final-annotation

```
1       In      _       _
2       prossimità      _       _
3       della   _       _
4       tornata _       _
5       elettorale      _       _
6       per     _       _
7       la      _       _
8       rielezione      _       _
9       delle   _       _
10      cariche _       _
11      di      _       _
12      assessori       _       _
13      alla    _       _
14      Regione _       _
15      Veneto  _       _
16      qualcuno        _       _
17      vuole   _       _
18      far-    _       1:ID
19      gli     _       _
20      le      _       1
21      scarpe  nsp     1
22      ?       _       _
```

Figure 4
Example of annotated data in parseme-tsv format

if a token starts an idiom which is the second VMWE in the current sentence).

In the case of nested, coordinated or overlapping VMWEs multiple codes are separated with a semicolon. Furthermore, in order to provide data usable as features in the shared task systems, also companion files in a format close to CoNLL-U [23] have been released. These companion files contain extra linguistic information, i.e., lemmas, POS-tags, morphological features, and syntactic dependencies.
Measuring inter-annotator agreement (IAA) is not a trivial task because of the challenges posed by VMWEs and described in the Introduction. Yet, for most languages, including Italian, the majority of the corpus has been annotated by a single annotator because of time and resource constraints. Thus, a small representative part of the corpus has been annotated by two annotators in order to calculate the IAA. The proposed IAA measures intend to assess different aspects, such as the resulting annotation, as well as the effort required in the annotation task and the guidelines and methodology applied. The available IAA results for the first edition of the PARSEME Shared Task, organized per-VMWE F-score (F_{unit}), estimated Cohen's K (K_{unit}) and finally standard K(K_{cat}) (Savary et al. 2017) scores are presented in Table 5.
To measure the unitising value [24] the MWE-based F-score (F_{unit}), as defined in Savary et al. (2017), has been calculated on one annotator with respect to the other considering the double-annotated data.
As noted by Markantonatou et al. (2018), measuring IAA, especially for Cohen's kappa (κ_{unit}), is not straightforward due to the lack of negative examples, namely spans formed of combination of a verb with other tokens (of any length) in a sentence for which both annotators agreed that they are not VMWEs. To reduce the bias in this measure with

23 http://universaldependencies.org/format.htm
24 Unitising is referred to the identification of the boundaries of a VMWE in the text;

Table 5
IAA scores for PARSEME-It VMWE corpus 1.0: #S, and #T show the number of sentences and tokens in the corpora used for measuring the IAA, respectively. $\#A_1$ and $\#A_2$ refer to the number of VMWE instances annotated by each of the annotators.

	#S	**#T**	**$\#A_1$**	**$\#A_2$**	**F_{unit}**	**κ_{unit}**	**κ_{cat}**
IT	2000	52639	336	316	0.417	0.331	0.78

reference to the F-score, the total number of possible VMWE candidates in the corpus has been assumed to be equivalent to the number of verbs, which is actually higher than the number of sentences and nevertheless estimated as the number of sentences plus the VMWE annotated at least by one annotator (Savary et al. 2017).
The standard κ (κ_{cat}) is applied to calculate the agreement on categorization, considering just the double-annotated VMWE spans. Italian, as other languages in the PARSEME annotation task, e.g., Spanish[25], shows low IAA scores, especially in unitising.
Table 6 shows the IAA scores for the second edition of PARSEME-It VMWE corpus.

Table 6
IAA scores for PARSEME-It VMWE corpus 1.1

	#S	**$\#A_1$**	**$\#A_2$**	**F_{span}**	**κ_{span}**	**κ_{cat}**
IT	1000	341	379	0.586	0.550	0.882

The IAA has been evaluated on a sample of 1,000 sentences, with A_1 and A_2 VMWEs annotated by each annotator. F_{span} is the F-measure between annotators, κ_{span} is the agreement on the annotation span and κ_{cat} is the agreement on the VMWE category (Ramisch et al. 2018). Although the IAA values increased in the second annotation campaign due to the presence of more fine-grained categories and better training of the annotators, these values are not so high, which can be explained by several reasons: (i) annotating some types of texts, i.e., Web texts in our corpus, are more difficult than annotating other types of texts, e.g., newspaper; (ii) double-annotated samples are quite small; (iii) guidelines and annotator training have to be improved. At any rate, these results call for a deep analysis of the issues arisen during the annotation, as presented in the following section.

7. Annotation Issues

In this section, we discuss the main annotation issues which emerged during the annotation finalized at assessing the IAA in the second edition of the Shared task. During this phase a set of 1,000 sentences was double-annotated by two different skilled annotators. The two annotators annotated almost the same number of VMWEs, namely ANNOTATOR1 341 VMWEs and ANNOTATOR2 379 VMWEs, but they completely agreed on the number of constituents and category only in 191 cases. VMWE annotation is a very hard task and disagreements occurred in different forms:

25 For IAA values for other languages, see Markantonatou et al. (2018).

1. Partial matches (labeled): this type refers to disagreements concerning the number of constituents of a VMWE labeled in the same way by both annotators;
2. Exact matches (unlabeled): this type refers to disagreements concerning the category of VMWE only;
3. Partial matches (unlabeled): this type refers to disagreements concerning the number of constituents and the category of a VMWE;
4. Single-annotated occurrences: this type refers to VMWEs annotated only by one annotator.

The disagreements will be discussed in the next subsections.

7.1 Partial Matches Labeled

A first source of disagreement is represented by the inclusion or exclusions of one or more constituents of VMWEs. Differences in annotation arise in relation to the judgment about the lexicalization of a component word of a VMWE, which might prove particularly difficult in presence of determiners, adjectives, pronouns/clitics, negations. In 25 cases different decisions were taken by the annotators on whether these words were part or not of VMWEs, resulting only in partial overlapping in the annotations, like in the examples provided below.

Inclusion/exclusion of determiners. The example provided in (1) refers to the VMWE *dare aiuto* (to help), which has been labeled as LVC.full by both annotators, but while ANNOTATOR1 identified the VMWE as *dare ... aiuto* ANNOTATOR2 included the determiner *un* as lexicalised constituent of the VMWE and therefore labeled *dare un aiuto*. In fact, it is possible to test whether a determiner is lexicalized by searching alternatives in dictionaries, corpora, or on the web. Borderline cases exist, in which alternatives are rare but possible, specially for LVCs and decomposable VIDs. The general rule, however, is that when alternatives are possible and the determiner varies, then it should not be included in the annotation.

1. `Source: PARSEME-It VMWE 2`
   Se sarà vero è una questione che dovranno risolverla tra loro e se qualcuno è a conoscenza dei fatti accaduti può **dare un aiuto** ad uno o all'altro contendente.
   (If it is true, they will have to solve the issue among themselves and if someone is aware of the events that have occurred, they can help one or the other contender.)

Inclusion/exclusion of adjectives. Another example of disagreement between annotators is given by the presence of an adjective which might be considered as part of a VMWE although it is not completely fixed, as in example (2) where both annotators identified the VID *porre in ... luce* but there was a different judgement with reference to the adjective *cattiva* as being part or not of the lexicalised constituents of the VMWE. This is due to the possibility to have alternative adjectives like *buona* as in *porre in buona luce* or *chiara* as in *porre in chiara luce*. The problem to be solved in this respect is to decide if the different adjectives convey a different meaning for the VMWE to be annotated.

2. Source: PARSEME-It VMWE 392
La stampa ha presentato la cosa in modo non corretto, ed alcuni commentatori l'avevano utilizzata, **ponendo in cattiva luce** l'immagine della Giolo, che si era limitata a fotografare per l'eventuale utilizzo in caso di ricorso.
(The press presented this incorrectly, and some commentators had used it, putting in a bad light the image of Giolo, who limited herself to take pictures for a possible use in case of appeal.)

Inclusion/exclusion of negations. Negations are usually also considered non lexicalized. However, this is not always the case and they might also represent a source of different judgments between annotators. For instance, the VID *non fare una cippa*, a substandard expression with the meaning of 'don't do anything' in example (3) presents a lexicalised negation which nevertheless causes some doubts in ANNOTATOR1 who does not annotate it as part of the VMWE.

3. Source: PARSEME-It VMWE 408
A me sembra, da esterna che segue da anni la manifestazione perchè a Rovigo quest'anno a mio parere **non hanno fatto una cippa** che stiate cercando di spremere un limone già secco.
(It seems to me, who has been following the event for years from outside since, in my opinion, they haven't done a bit in Rovigo this year, that you are trying to squeeze an already dry lemon.)

Inclusion/exclusion of clitics. Clitics also challenge very often judgments as to whether they are part or not of VMWEs like in *fare le spese*. In example (4) only ANNOTATOR2 annotated the non-reflexive clitic *-ne* as a constituent of a VID, considering it as a fixed element of the VMWE.

4. Source: PARSEME-It VMWE 492
è tutto uno scaricabarile... e a **farne le spese** sono i ragazzi.
(it's all passing the buck ... and the boys are the ones who pay for it.)

Inclusion/exclusion of pronouns. Pronouns, indeed are also usually non-lexicalised since they can vary, but example (5) caused another disagreement as to whether the pronoun is a constituent or not of a VMWE. Here the judgment of the annotator that included the personal dative pronoun *ti* in the annotation of the VMWE *stare bene* probably is based on the idea that the meaning of the VMWE *stare bene a qualcuno* (to look good on someone) is different from the meaning of *stare bene* (to feel well). In this case, the presence of the pronoun conveys a completely different meaning although it is not invariable as other personal pronouns are equally acceptable, e.g. *(mi/ti/gli/...) sta bene*.

5. Source: PARSEME-It VMWE 966
Certo che **ti sta** proprio **bene**... è questa la sorpresa?
(It looks good on you ... is this the surprise?)

Mistakes in annotations. In the category of partial matches labeled there are also 4 mistakes, such as annotation of single words instead of multiwords, or un-annotated

elements of a VMWE. For instance, in example (6) ANNOTATOR1 did not annotate the verb of the VID *mettete*, while ANNOTATOR2 annotated it.

6. `Source: PARSEME-It VMWE 646`
Perché non ne abbiamo già abbastanza di fastidi tra Spinello e Barbujani e **vi ci mettete** anche voi?
(Don't we have enough of annoyances between Spinello and Barbujani and do you contribute too?)

7.2 Exact Matches Unlabeled

In this case, annotators identify the same constituents but disagree on the category of VMWEs. The disagreements (18 cases) mainly concern LVCs (full and cause) and VPCs (full and semi): these categories are very fine-grained and pose some problems in the assessment of the grade of non- compositionality. Another frequent source of disagreement concerns different decisions as to whether a VMWE belongs to the VID or LVC category(both full and cause). Disagreements concerning exact matches were eliminated in version 1.2 of the corpus [26].

VPC. As already mentioned, in fully non-compositional VPC (VPC.full) the change in the meaning of the verb goes significantly beyond adding the meaning of the particle: like for *buttare giù* with the meaning of *to swallow*. In semi-non-compositional VPCs (VPC.semi), the particle adds a partly predictable but non-spatial meaning to verb: like in *lasciare dietro* with the meaning of *to leave behind*. The LVC *mettere insieme* causes some uncertainties as to whether it is a VPC.full (ANNOTATOR1) or a VPC.semi (ANNOTATOR2).

7. `Source: PARSEME-It VMWE 7`
... ringrazio il sindaco Barbujani e la giunta che ha permesso di **mettere insieme** un programma di tutto rispetto.
(I thank Mayor Barbujani and the council that made it possible to put together a very respectable program.)

LVC. The verb is "light" in that it contributes to the meaning of the whole only by bearing morphological features: person, number, tense, mood, as well as morphological aspects. This implies that the syntactic subject of the verb is the semantic argument of the noun [27]. In this case, we annotate the construction as LVC.full like in *fare una presentazione* (to make a presentation). If the verb is "causative" in that it indicates that the subject of the verb is the cause or source of the event or state expressed by the noun, the VMWE should be annotated as LVC.cause like in *dare le vertigini* (to make dizzy). In example (8) annotators do not agree on the LVC type of the verb *dare fiducia* and ANNOTATOR1 labels it as LVC.cause while ANNOTATOR2 as LVC.full.

8. `Source: PARSEME-It VMWE 810`

26 https://gitlab.com/parseme/sharedtask-data/-/tree/master/1.2/IT
27 https://parsemefr.lislab.fr/parsemestguidelines/1.1/?page=050_Cross-lingual_tests/020_Light-verb_constructions__LB_LVC_RB_

La nostra volontà la vogliamo portare in Consiglio Comunale approvando il PAT, che è a portata di mano, che è in dirittura d'arrivo e servirà **dare fiducia** ai cittadini.
(We want to bring our will to the City Council by approving the PAT, which is close at hand, which is in the home stretch and will serve to trust citizens.)

Disagreement about VID and LVC. A frequent disagreement between the annotators concerned the VID and LVC categories, like example (9) where the VMWE *fare la parte* was annotated as a VID by ANNOTATOR1 and as a LVC by ANNOTATOR2. This uncertainty may be due to different judgments given to the tests applied in the decision process. In particular the annotators might have taken different decisions with respect to some tests concerning VIDs, like Test VID.2 - [LEX] - Lexical inflexibility: this test requires to assess whether the regular replacement of one of the components by related words taken from a relatively large semantic class leads to ungrammaticality or to an unexpected change in meaning, for instance in this case whether the replacement of the verb *fare* with *sostenere* or of the determiner *la* with the indefinite article *una* leads to different meanings. In case of a negative answer, annotators should have taken Test VID.3 - [MORPH] - Morphological inflexibility which requires to assess whether regular morphological change that would normally be allowed by general grammar rules leads to ungrammaticality or to an unexpected change in meaning, for instance, whether *fare le parti* has a different meaning with respect to *fare la parte*. Therefore, while ANNOTATOR1 answered positively to one of the abovementioned tests, ANNOTATOR2 answered negatively to them and answered positively to one of the tests for LVCs[28].

9. `Source: PARSEME-It VMWE 425`
Lasciate lavorare la maggioranza e lasciate l'opposizione **fare la parte** che gli compete.
(Let the majority work and let the opposition do its part.)

7.3 Partial Matches unlabeled

The only case of partial match unlabeled concerns a different interpretation of the VMWE both in terms of the number of constituents and category attribution. The example (10) presents the VMWE *buttarsi (nella calca)* which was labeled by ANNOTATOR1 as *buttar-si in la calca* (VID) and by ANNOTATOR2 *buttar-si* (IRV).

10. `Source: PARSEME-It VMWE 864`
Chi è rimasto nei pressi della propria città approfittandone per sistemare casa, alzandosi la mattina tardi, passeggiando per il corso attendendo il venerdì sera per **buttarsi nella calca** del divertimento...
(Those who stayed close to their city taking advantage of it to settle home, getting up late in the morning, walking along the street waiting for Friday evening to mix in the crowd of fun ...)

28 https://parsemefr.lis-lab.fr/parseme-st-guidelines/1.1/?page=lvc#decision-tree-lvc

7.4 Single-Annotated Occurrences

The main problems of disagreement lie in the high number of VMWEs annotated by only one annotator: 250 cases split in 106 for ANNOTATOR1 and 144 for ANNOTATOR2 (Table 7).

Table 7
Single-annotated occurrences for each category

IAV	VID	MVC	LVC.full	LVC.cause	VPC.full	VPC.semi	IRV	LS.ICV
28	66	5	72	10	11	1	47	10

From these figures, it emerges very clearly that the most problematic VMWE category is represented by LVCs. One possible reason is that the verbs of LVCs are very common ones such as *fare* (*fare ricorsi, fare errori*), *dare* (*dare allucinazioni, dare informazioni*), *prendere* (*prendere visione*), *avere* (*avere difficoltà, avere esperienza*) and since these verbs share the same meaning with other lexical constructions which are not LVCs, annotators may not identify them as such. For instance, the verb *avere* does not change its meaning from *avere una sedia* (non-VMWE) to *avere difficoltà* (VMWE). Besides, it is clear from the annotations that sometimes meaning-preserving variants of a (candidate) VMWE such as verbal expressions with analytical tenses and modals, like in *hanno preso una decisione*, nominal groups (headed by nominal complements from the prototypical VMWEs) with relative clauses (e.g., *i cuori che abbiamo spezzato*), non-finite verbal clauses (e.g., *decisioni prese precedentemente*), diathesis alternation (*decisioni importanti sono state prese*) may cause problems in the identification of VMWEs. Also, VID seems to be quite problematic (66 cases): our intuition about this type of disagreement is that some VIDs are not considered sufficiently established in the common vocabulary such as *mettere su pignatta* but also because it is often challenging to distinguish VIDs when only one dependent of the head verb is lexicalized or when they occur in variants which, as already stated, might cause overlooking VMWEs and inattentions in the annotation.

8. Linguistic Observations

In this section, we discuss some linguistic observations on IRVs and VIDs, which are very productive categories, and a comparison between LVC and IAV, as their categorization may rise some borderline cases. Even though it is an interesting phenomenon, we do not offer a deep analysis on the status of VPCs in Italian since the number of occurrences in the PARSEME-It VMWE corpus is not so high and, therefore, not representative. In fact, as a Romance language, Italian was expected not to exhibit VPCs, but several dozens of VPC annotations do occur in the corpus, e.g., *volata via* (lit. 'flew part' $\rightarrow$ slipped away), *tira fuori* (lit. 'he pulls part' $\rightarrow$ he shows), or *va avanti* (lit. 'go part' $\rightarrow$ go on). This shows the possibly ambiguous status of the element co-occurring with the verb, that is, in previous examples, *via* (by/away), *avanti* (on/forward), *fuori* (out/outside), which can be either adverbs or particles, triggering the VID or the VPC category, respectively. These constructions require to be examined more closely, thus a higher number of occurrences in the corpus is required.

8.1 Very productive VMWEs: IRVs and VID

As described in Monti et al. (2018), IRVs and VID represent very productive categories in Italian which pose some classification issues due to their specific characteristics.

With reference to **IRVs**, the first source of ambiguity in the annotation process is the presence of the clitic pronoun *si* in that in Italian it may be used in three types of different constructions: i) reflexive, ii) impersonal, iii) inherent.

In order to exclude from the annotation reflexive verbs as IRVs, we consider that in reflexive constructions, the clitic pronoun *si* marks the reflexive or reciprocal construction, that is, the clitic plays the role of *self* in English and can be paraphrased by means of either an anaphoric expression which stands for *se stesso* (oneself) or a mutual expression which refers to *gli uni e gli altri* (these and those).

To prevent the annotation of impersonal constructions, not belonging to the IRV class, we observe that in these cases the clitic *si* co-occurs with either an intransitive verb or a transitive verb in third person singular. In these occurrences, both classes, originally presenting one or two arguments, reduce their usual number of valency slots to zero, namely they present an empty subject slot, as they convey an absolute and universal meaning expressed by a generic and underspecified subject, e.g., *si muore* (lit. *dies itself $\rightarrow$ dying), *si pensa* (lit. *thinks itself $\rightarrow$ thinking).

Furthermore, as already stated previously, inherent uses of the pronoun *si* are annotated as IRVs, if the verb without the clitic does not exist, e.g., *vergognarsi* (to feel ashamed), or if the verb without the clitic does exist and conveys a very different meaning, e.g., *raffreddarsi* (to get a cold), *raffreddare* (to cool down).

Another relevant aspect to consider in the classification of IRVs is the presence of an implicit thematic role due to the fact that the action includes two different entities with different thematic properties but with the same reference, e.g., in *guardarsi* (to look at oneself) the clitic signals the presence of coreference between the first argument and the second one.

Among sources of mis-classification of IRVs, we notice that the presence of unaccusative constructions (Perlmutter 1978) may generate ambiguity. In fact, in these occurrences, formed through a pseudo-reflexive construction, the clitic, usually representing an overt marker of reduced transitivity, e.g., *seder**si*** (to sit down), is not marked by the accusative case. Unaccusative verbs may be distinguished by applying both semantic and syntactic criteria. Semantically, unaccusative verbs are characterized in that their meaning stands for a change of state, in other words these verbs express telicity, as *sedersi*. From a syntactic point of view, these verbs select a specific temporal auxiliary verb, that is they combine with *be*, while unergative constructions use the verb *have*.

In some cases, IRVs occur in idiomatic constructions and their meaning is affected by the presence of new elements, such as in *guardarsi bene da* (to be careful not to). Consequently the annotation of such occurrences is subjected to the evaluation of characteristics related to VID, as the low variability, the presence of semantic non-compositional meaning, and the literal-idiomatic ambiguity.

In the **VID** class, the non-compositionality property is prototypical such as in *battersi all'ultimo sangue* (lit. 'to fight till the last blood') which means *to fight to the last*. Despite their meaning is opaque, sometimes VIDs may have both a literal and idiomatic meaning and the boundaries between them are difficult to trace. For instance, *avere gli occhi bendati* (lit. 'to have the eyes covered') has both a literal meaning and an idiomatic one and in this latter case it should be translated in English as *to be blindfold*.

According to Vietri (2014c), it is possible to classify ordinary-verb VIDs, namely VIDs which present a semantically full verb, on the basis of their definitional structure,

identified by means of the arguments required by the operators. In the case of VID, the operator consists of the verb and the fixed element(s), while the argument may be the subject and/or a free complement. The fixed dependent can be of different types:

- Subject, e.g., *un uccellino mi ha detto* (a bird told me)
- Direct object, e.g., *tirare le cuoia* (kick the bucket)
- Circumstantial or adverbial complement, e.g., *prendere qualcosa con le pinze* (to take something with a pinch of salt)

VIDs can be formed also by constructions based on the use of support verbs, namely *avere* (to have), e.g., *avere fegato* (lit. 'to have leaver' → to have guts) *essere* (to be), e.g., *essere a cavallo* (to be golden) and *fare* (to make), e.g., *fare lo gnorri* (to play fool). The main difference between this class of VID and the one formed by ordinary verbs is that support verbs are semantically empty, and, for this reason, this class of VID presents a high degree of lexical and syntactic variability. This type of variability is retrievable in aspectual variants, production of causative constructions, possible deletion of the support verb which causes complex nominalizations (Vietri 2014a).

8.2 Borderline cases: LVC and IAV compared

During the annotation process, other borderline cases were identified in two categories, namely LVC and IAV[29], used in the second edition of the shared task.

As previously stated, the former, already annotated in the first edition of the task, has been modified to account for a more fine-grained distinction, i.e., it has been split into LVC.full and LVC.cause.

On one hand, **LVC.full** accounts for occurrences in which the verb contributes to the MWE meaning in that it bears only morphological features, namely person, number, tense, mood, as well as morphological aspect. This implies that the syntactic subject of the verb is the semantic argument of the noun. Such a definition of LVC is different from the one usually proposed by many authors (Hopper and Traugott 2003; Hacker 1958; Hook 1991, 1993) for two main aspects. At first, we do not include aspectual support verbs, unless the aspect is morphological. We do not consider aspectual verbs contributing to a change of the MWE meaning, (e.g., *iniziare* → to start) since, despite the fact they are very productive, they do not form interesting VMWEs (Savary et al. 2017). Therefore, we annotate occurrences in which a predicative noun, e.g., *passeggiata* (walk), co-occurs with a light verb, e.g., *fare*, such in *fare una passeggiata* (have a walk), nevertheless discarding occurrences with aspectual verbs, e.g., *iniziare una passeggiata* (to start a walk). Then, in addition to the standard definition, we take into account also verbs presenting a light semantics per se, which are not considered bleached support verbs. In this perspective, the occurrence *commettere un suicidio* (to commit a suicide) passes all tests and may be accounted as an LVC.full in that it preserves its meaning defined by the presence of any negatively charged achievement noun, e.g. suicide, crime, fraud, felony.

29 This section is partially based on the analysis presented by Caruso in Monti et al. (2018).

On the other hand, **LVC.cause** constructions, expected to be less idiomatic than other VMWEs, can be understood as complex predicates with a causal support verb[30]. In these occurrences, the verb is considered causative when the subject of V is the cause or the main source of the event or state expressed by the noun, e.g., *dare il mal di testa* (to give a headache). LVC.cause constructions may involve:

- verbs that are typically used to express the cause of predicative nouns in general (e.g., *cause, provoke*)
- verbs that are only used to express the cause of particular predicative nouns (e.g., *grant* in *to grant a right*).

Some new tests have been added to account for these subcategories, which heavily rely on the notion of semantic arguments. These tests aim at distinguishing cases in which: (i) the noun is predicative; (ii) the verb's subject is a noun's semantic argument; (iii) the verb presents a light semantics; (iv) the verb reduction is applicable; (v) the verb's subject is the noun's cause.

As already described, **IAVs** are a special optional and experimental category, and correspond to what is also sometimes called in English prepositional verbs, as they consist of a verb or VMWE and an idiomatic selected preposition or postposition. Since in some cases the idiomatic adpositional valency, namely when the co-occurrence of a verb with an adposition opens a slot for an argument, may be mistaken with verb-particle constructions, a language-specific test, mainly concerning English and German, has been provided. Generally speaking, these two phenomena can be distinguished by analyzing the adposition behaviour. If it can occur after the object, e.g., *to wake somebody up*, then the adposition is a particle and the group can not be categorized as IAV. If the adposition cannot occur after the object, as in **to come something across*, then the MWE belongs to the IAV category.

During the annotation trial, the IAV category has proved to be advantageous to cover a rich inventory of VMWEs in Italian, but some issues have also emerged, particularly with respect to the LVC verbs, which also account for combinations of verbs plus prepositions. Prototypical examples of IAV collected so far include the following:

- *Tendere a* + N (to be inclined to something), base form *tendere* (to stretch), e.g., *Maria tende alla depressione* (Maria tends to be depressed);
- *Tendere a* + V (to be inclined to something), e.g., *Maria tende a dimagrire* (Maria tends to loose weight);
- *Puntare su* + N (to bet), base form *puntare* (to stick), e.g., *puntare su qualcuno/qualcosa*.

These examples exhibit clear semantic changes from the non-adpositional base form of the verb; moreover, the preposition cannot be omitted in questions, thus proving to be part of the verb, as in the following example.

11. Maria **tende** sempre **ad** esagerare. (*Maria always tends to overstate*)
A cosa **tende**, scusa? *(What does she tend to?)*

30 https://parsemefr.lis-lab.fr/parseme-st-guidelines/1.1/?page=050_Cross-lingual_tests/020_Light-verb_constructions__LB_LVC_RB_

Less prototypical IAV examples include verb instances exhibiting semantic changes pivoted by the arguments they combine with, like *andare in* (both *to go to* and *to become*), or *sapere di* (*to smell* and *to know about*). The type of semantic interaction at stake, called *co-composition* in the Generative Lexicon[31], is realized when "the complements carry information which acts on the governing verb, essentially taking the verb as an argument and shifting its event type" (Pustejovsky 1995). For instance, *andare in* denotes directed motion when combined with proper or common place nouns like in *andare in città/montagna/America*, (to go to the city/mountain/America); or the medium of motion, when combined with vehicles names, like in *vado in bici/Ferrari* (I ride my bike/drive my Ferrari). However, with nouns denoting *states*, like *andare in estasi* (to become absorbed) or *andare in panico* (to panic), the verb acquires the aspectual meaning of *to go into the state X*, and cannot be classified as an LVC. With names referring to events, instead, like *andare in soccorso* (lit. 'to go in assistance'), the original spatial semantics bleaches by interacting with the name meaning: actually *to go into the event X* denotes the action expressed by the predicative name and can be classified as an LVC. Therefore, a more fine-grained analysis is needed in order to annotate these categories appropriately, and capture significant semantic differences. As a counter-example, giving evidence to the broad coverage of the IAV class, one can refer to *portare a* (carry/bring to), because its causative semantics, derived from an original spatial meaning, remains unchanged in different lexical and syntactic contexts. Both with nouns denoting a state (e.g., *portare alla follia*, lit. 'to bring someone to madness'), with those referring to events (*portare a ebollizione*, lit. 'to bring something to boiling point'), and with full-sentence arguments (*portare a conoscere*, lit. 'to bring someone know something') *portare a* preserves its causative meaning.

9. Conclusion and future work

In this paper, we described a linguist resource of Italian VMWEs, developed within the PARSEME Shared Task on Automatic Identification of VMWEs. To the best of our knowledge, PARSEME-It represents the first annotated corpus for Italian VMWEs. Firstly, we introduced current works focused on MWE processing from different perspectives, i.e., linguistic studies and NLP applications. Then, we described aims and methodologies used within the PARSEME Cost Action to define the research objects and to identify such linguistic phenomena. Subsequently, we described the development of the PARSEME-It VMWE corpus and the VMWE categories we took into account within the framework of the PARSEME Shared Task on Automatic Identification of VMWEs (Savary et al. 2017; Ramisch et al. 2018).
Then, we discussed the annotation guidelines together with the identification tests and the category decision trees applied to identify and classify VMWEs. The PARSEME-It VMWE corpus is based on a selection of texts, formed by approx. 16,000 sentences (corresponding to 430,789 tokens) taken from the *PAISÀ* corpus of Italian web texts. The annotation process together with the IAA is presented. A deep analysis of the issues arisen during the double-annotation task shows the disagreement cases in IAA scores. Several sources of disagreement have been identified, namely partial matches labeled, exact matches unlabeled, partial matches unlabeled, and finally VMWE annotations by only one annotator. Yet, among the annotated occurrences, we proposed an analysis of productive categories, i.e., IRVs and VIDs, and a comparison of LVC and IAV categories.

31 Co-composition has been called *accommodation* in more recent works (Pustejovsky 2013).

Due to the high complexity of this type of phraseological units, we consider this work an initial contribution for elaborating an Italian universal terminology of VMWEs, which could ease the challenge of MWE automatic processing, in particular verbal ones.
Furthermore, the analysis of these linguistic phenomena could represent the foundation for semantic representation, suitable to encompass cross-lingual comparisons and applications.
Future work includes the extension of the current corpus and a fine-grained linguistic analysis of the annotation in order to contribute to the description of these phenomena, increasing the quality of multilingual dictionaries and allowing their full integration into emerging language technologies (LTs). These technologies are based on a semantic formalized representation, which encodes several levels of linguistic information, suitable to guarantee the interoperability among resources from different sources and languages.
The properties of verbal multiword expressions in Italian may contribute to improving their semantic representation according to W3C standards used in current LTs, namely the OntoLex Lemon model[32]. This model aims at providing a rich linguistic grounding for ontologies, including the representation of morphological and syntactic properties of lexical entries as well as the syntax-semantics interface, i.e., the meaning of these lexical entries with respect to an ontology or vocabulary (McCrae et al. 2017). The use of this type of formalization to describe linguistic data and resources represents a straight way to contribute to the development of a Linguistic Linked Open Data (LLOD) cloud[33], creating, sharing, and (re-)using language resources in accordance with Linked Data principles (Bizer, Heath, and Berners-Lee 2008).

Acknowledgments

This work has been partially supported by Programma Operativo Nazionale Ricerca e Innovazione 2014-2020 - Fondo Sociale Europeo, Azione I.2 "Attrazione e Mobilità Internazionale dei Ricercatori" Avviso D.D. n 407 del 27/02/2018 and by the IC1207 PARSEME COST action (2013-2017).
We are particularly grateful to Federico Sangati who always supported the annotation team and actively took part in the planning and the implementation of the project. Finally, our thanks go also to all the Italian annotators, which took part in editions 1.0 and 1.1 of the PARSEME shared task on verbal MWE identification, namely Valeria Caruso, Manuela Cherchi, Anna De Santis, Antonio Pascucci, Annalisa Raffone, and Anna Riccio.
Authorship contribution is as follows: Johanna Monti is author of sections 1, 3, 4, 5 and 7. Maria Pia di Buono is author of sections 2, 6, 8 and 9. Abstract is in common.

References

Alba-Salas, Josep. 2002. *Light Verb Constructions in Romance: A syntactic analysis*. Ph.D. thesis, Cornell University, NYC, New York.
Alba-Salas, Josep. 2004. Fare light verb constructions and Italian causatives: Understanding the differences. *Italian Journal of Linguistics*, 16(2):283.
Baldwin, Timothy. 2006. Compositionality and multiword expressions: six of one, half a dozen of the other? In *Proceedings of the Workshop on Multiword Expressions: Identifying and Exploiting Underlying Properties*, Sydney, July.
Baldwin, Timothy and Su Nam Kim. 2010. Multiword Expressions. In Nitin Indurkhya and Fred J. Damerau, editors, *Handbook of Natural Language Processing, Second Edition*. CRCPress, Boca Raton, USA, pages 267–292.

32 https://www.w3.org/community/ontolex/wiki/Final_Model_Specification
33 https://en.wikipedia.org/wiki/Linguistic_Linked_Open_Data

Barreiro, Anabela, Johanna Monti, Brigitte Orliac, Susanne Preuß, Kutz Arrieta, Wang Ling, Fernando Batista, Isabel Trancoso, et al. 2014. Linguistic evaluation of support verb constructions by openlogos and google translate. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*, pages 35–40, Reykjavik, Iceland, May.

Berruto, Gaetano. 1987. *Sociolinguistica dell'italiano contemporaneo*, volume 33. Carocci, Roma.

Bizer, Christian, Tom Heath, and Tim Berners-Lee. 2008. Linked Data: Principles and State of the Art. In *17th International World Wide Web Conference*, volume 1, page 40, Beijing, China, April.

Butt, Miriam. 2010. The light verb jungle: Still hacking away. In Mengistu Amberber, Brett Baker, and Mark Harvey, editors, *Complex predicates in cross-linguistic perspective*. Cambridge University Press Cambridge, MA, Cambridge, pages 48–78.

Cap, Fabienne, Manju Nirmal, Marion Weller, and Sabine Schulte Im Walde. 2015. How to account for idiomatic German support verb constructions in statistical machine translation. In *Proceedings of the 11th Workshop on Multiword Expressions*, pages 19–28, Denver, Colorado, June.

Carstea-Romascanu, Mihaela. 1977. I tipi di verbi riflessivi in italiano. *Revue Roumaine de Linguistique Bucuresti*, 22(2):125–130.

Cicalese, Anna. 1999. Le estensioni di verbo supporto. uno studio introduttivo. *Studi italiani di linguistica teorica ed applicata*, 28(3):447–485.

Cicalese, Anna, Emilio D'Agostino, Alberto Maria Langella, and Ilaria Villari. 2016. Els verbs locatius com a variants de verbs de suport. *Quaderns d'Italià*, 21(21):153–166.

Cinque, Guglielmo. 1988. On si constructions and the theory of arb. *Linguistic inquiry*, 19(4):521–581.

Constant, Mathieu, Gülşen Eryiğit, Johanna Monti, Lonneke Van Der Plas, Carlos Ramisch, Michael Rosner, and Amalia Todirascu. 2017. Multiword expression processing: A survey. *Computational Linguistics*, 43(4):837–892.

Constant, Matthieu, Anthony Sigogne, and Patrick Watrin. 2012. Discriminative Strategies to Integrate Multiword Expression Recognition and Parsing. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 204–212, Jeju, Republic of Korea. Association for Computational Linguistics.

Copestake, Ann. 2003. Compounds revisited. In *Proceedings of the 2nd International Workshop on Generative Approaches to the Lexicon*, pages 129–154, Geneva, Switzerland, June.

Cordin, Patrizia. 2001. I pronomi riflessivi. In Lorenzo Renzi, Giampaolo Salvi, and Anna Cardinaletti, editors, *Grande grammatica italiana di consultazione*, volume 1. Il Mulino, Bologna, pages 607–17.

D'Agostino, Emilio and Annibale Elia. 1998. Il significato delle frasi: un continuum dalle frasi semplici alle forme polirematiche. In Federico Albano Leoni, Daniele Gambarara, Stefano Gensini, Franco Lo Piparo, and Raffaele Simone, editors, *Ai limiti del linguaggio*. Laterza, Bari, pages 287–310.

de Caseli, Helena Medeiros, Carlos Ramisch, Maria das Graças Volpe Nunes, and Aline Villavicencio. 2010. Alignment-based extraction of multiword expressions. *Language resources and evaluation*, 44(1-2):59–77.

De Mauro, Tullio. 2000. *Grande dizionario italiano dell'uso (GRADIT)*. Utet, Torino.

Gagné, Christina L and Thomas L Spalding. 2009. Constituent integration during the processing of compound words: Does it involve the use of relational structures? *Journal of Memory and Language*, 60(1):20–35.

Gibbs, Raymond W, Nandini P Nayak, John L Bolton, and Melissa E Keppel. 1989. Speakers' assumptions about the lexical flexibility of idioms. *Memory & cognition*, 17(1):58–68.

Hacker, Paul. 1958. *Zur Funktion Einiger Hilfsverben im Modernen Hindi*. Verlag der Akademie der Wissenschaften und der Literatur in Mainz, München.

Hook, Peter Edwin. 1991. The Emergence of Perfective Aspect in Indo-Aryan languages. *Approaches to grammaticalization*, 2:59–89.

Hook, Peter Edwin. 1993. Aspectogenesis and the Compound Verb in Indo-Aryan. In Manindra K. Verma, editor, *Complex predicates in South Asian languages*. Manohar Publishers & Distributors, New Delhi, pages 97–113.

Hopper, Paul J and Elizabeth Closs Traugott. 2003. *Grammaticalization*. Cambridge University Press, Cambridge.

Iacobini, Claudio and Francesca Masini. 2005. Verb-particle constructions and prefixed verbs in Italian: typology, diachrony and semantics. In *Mediterranean Morphology Meetings*, volume 5, pages 157–184, Fréjus, France, September.

Im Walde, Sabine Schulte, Stefan Müller, and Stefan Roller. 2013. Exploring Vector Space Models to Predict the Compositionality of German Noun-Noun Compounds. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 255–265, Atlanta, Georgia, June.

Jackendoff, Ray. 1997. *The architecture of the language faculty*. MIT Press, Cambridge.

Jezek, Elisabetta. 2004. Types et degrés de verbes supports en italien. *Lingvisticæ Investigationes*, 27(2):185–201.

Karimi-Doostan, Gholamhossein. 1997. *Light Verb Constructions in Persian*. Ph.D. thesis, University of Essex, Colchester, Essex, UK.

Kordoni, Valia and Iliana Simova. 2014. Multiword expressions in machine translation. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*, pages 1208–1211, Reykjavik, Iceland, May.

La Fauci, Nunzio. 1980. Aspects du mouvement de wh, verbes supports, double analyse, complétives au subjonctif en italien: pour une description compacte. *Lingvisticae Investigationes*, 4(2):293–341.

Laranjeira, Bruno, Viviane Pereira Moreira, Aline Villavicencio, Carlos Ramisch, and Maria José Bocorny Finatto. 2014. Comparing the quality of focused crawlers and of the translation resources obtained from them. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*, pages 3572–3578, Reykjavik, Iceland, May.

Lenci, Alessandro, Gianluca Lebani, Sara Castagnoli, Francesca Masini, and Malvina Nissim. 2014. Sympathy: Towards a comprehensive approach to the extraction of Italian word combinations. In *Proceedings of the First Italian Conference on Computational Linguistics, CLiC-it*, pages 234–238, Pisa, December. Pisa University Press.

Lyding, Verena, Egon Stemle, Claudia Borghetti, Marco Brunello, Sara Castagnoli, Felice Dell'Orletta, Henrik Dittmann, Alessandro Lenci, and Vito Pirrelli. 2014. The PAISÀ corpus of Italian web texts. In *Proceedings of the 9th Web as Corpus Workshop (WaC-9)*, pages 36–43, Gothenburg, Sweden, April.

Lyse, Gunn Inger and Gisle Andersen. 2012. Collocations and statistical analysis of n-grams. In Gisle Andersen, editor, *Exploring Newspaper Language: Using the web to create and investigate a large corpus of modern Norwegian*, volume 49. John Benjamins Publishing, Amsterdam/New York, pages 79–109.

Markantonatou, Stella, Carlos Ramisch, Agata Savary, and Veronika Vincze, editors. 2017. *Proceedings of the 13th Workshop on Multiword Expressions (MWE 2017)*, Valencia, Spain, April. Association for Computational Linguistics.

Markantonatou, Stella, Carlos Ramisch, Agata Savary, and Veronika Vincze. 2018. *Multiword expressions at length and in depth: Extended papers from the MWE 2017 workshop*. Language Science Press, Berlin, Germany.

Masini, Francesca. 2005. Multi-word expressions between syntax and the lexicon: The case of Italian verb-particle constructions. *SKY Journal of Linguistics*, 18(2005):145–173.

Masini, Francesca. 2015. Idiomatic verb-clitic constructions: Lexicalization and productivity. In *Proceedings of Mediterranean Morphology Meetings*, volume 9, pages 88–104, Haifa, Israel, September.

Mateu, Jaume and Gemma Rigau. 2010. Verb-particle constructions in Romance: A lexical-syntactic account. *Probus*, 22(2):241–269.

McCrae, John P, Julia Bosque-Gil, Jorge Gracia, Paul Buitelaar, and Philipp Cimiano. 2017. The Ontolex-Lemon model: Development and Applications. In *Proceedings of eLex 2017 conference*, pages 19–21, Leiden, Netherlands, September.

Mitkov, Ruslan, Johanna Monti, Gloria Corpas Pastor, and Violeta Seretan. 2018. *Multiword Units in Machine Translation and Translation Technology*, volume 341. John Benjamins Publishing Company, Amsterdam/New York.

Monti, Johanna, Anabela Barreiro, Brigitte Orliac, and Fernando Batista. 2013. When Multiwords Go Bad in Machine Translation. In *Machine Translation Summit XIV*, pages 26–33, Nice, France, September. The European Association for Machine Translation.

Monti, Johanna, Valeria Caruso, and Maria Pia di Buono. 2018. PARSEME-It-Issues in verbal Multiword Expressions identification and classification. In *Proceedings of the Fifth Italian Conference on Computational Linguistics (CLiC-it 2018)*, volume 2253, Torino, Italy, December. Accademia University Press.

Monti, Johanna, Silvio Cordeiro, Carlos Ramisch, Federico Sangati, Agata Savary, and Veronika Vincze. 2018. Advances in Multiword Expression Identification for the Italian language: The

PARSEME shared task edition 1.1. In *Proceedings of the Fifth Italian Conference on Computational Linguistics (CLiC-it 2018)*, volume 2253, Torino, Italy, December. Accademia University Press.

Monti, Johanna, Maria Pia di Buono, and Federico Sangati. 2017. PARSEME-It corpus: An annotated Corpus of Verbal Multiword Expressions in Italian. In *Fourth Italian Conference on Computational Linguistics-CLiC-it 2017*, pages 228–233, Rome, Italy, December. Accademia University Press.

Monti, Johanna, Ruslan Mitkov, Gloria Corpas Pastor, and Violeta Seretan, editors. 2013. *Workshop Proceedings. Multi-Word Units in Machine Translation and Translation Technologies. MUMTTT 2013*, Switzerland, September. Tradulex.

Monti, Johanna, Mitkov Ruslan, Seretan Violeta, and Gloria Corpas Pastor. 2018. *Proceedings of the 3rd Workshop on Multi-Word Units in Machine Translation and Translation Technology (MUMTTT 2017)*. Tradulex, Switzerland, November.

Newman, David, Nagendra Koilada, Jey Han Lau, and Timothy Baldwin. 2012. Bayesian Text Segmentation for Index term Identification and Keyphrase Extraction. In *Proceedings of COLING 2012, the 24th International Conference on Computational Linguistics: Technical Papers*, pages 2077–2092, Mumbai, India, December.

Ninio, Anat. 2011. *Syntactic development, its input and output*. Oxford University Press, Oxford, UK.

Nissim, Malvina, Sara Castagnoli, and Francesca Masini. 2014. Extracting MWEs from Italian corpora: A case study for refining the pos-pattern methodology. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics - EACL 2014*, pages 57–61, Gothenburg, Sweden, April.

Nunberg, Geoffrey, Ivan A Sag, and Thomas Wasow. 1994. Idioms. *Language*, 70(3):491–538.

Ó Séaghdha, Diarmuid. 2008. Learning compound noun semantics. Technical report, University of Cambridge, Computer Laboratory, Cambridge.

Pal, Santanu, Sudip Kumar Naskar, and Sivaji Bandyopadhyay. 2013. A hybrid word alignment model for phrase-based statistical machine translation. In *Proceedings of the Second Workshop on Hybrid Approaches to Translation*, pages 94–101, Sofia, Bulgaria, August.

Pastor, Gloria Corpas, Ruslan Mitkov, Maria Kunilovskaya, and María Araceli Losey León, editors. 2019. *Computational and Corpus-based Phraseology Proceedings of the Third International Conference EUROPHRAS 2019 (short papers, posters and MUMTTT workshop contributions)*, Switzerland, September. Tradulex.

Pastor, Gloria Corpas, Johanna Monti, Violeta Seretan, and Ruslan Mitkov, editors. 2015. *Workshop Proceedings. Multi-Word Units in Machine Translation and Translation Technologies. MUMTTT 2015*, Switzerland, July. Tradulex.

Perlmutter, David M. 1978. Impersonal Passives and the Unaccusative Hypothesis. In *Proceedings of the 4th Annual Meeting of the Berkeley Linguistics Society*, volume 4, pages 157–190, Berkeley, California, February.

Pescarini, Diego. 2015. Costruzioni con si: una classificazione razionale? *Grammatica e Didattica*, pages 15–32.

Pustejovsky, James. 1995. *The generative lexicon*. MIT Press, Cambridge.

Pustejovsky, James. 2013. Type theory and lexical decomposition. In James Pustejovsky, Pierrette Bouillon, Hitoshi Isahara, Kyoko Kanzaki, and Chungmin Lee, editors, *Advances in generative lexicon theory*. Springer, Berlin, Germany, pages 9–38.

Quaglia, Stefano and Andreas Trotzke. 2017. Italian verb particles and clausal positions. In *Proceedings of The 31st annual meeting Israel Association for Theoretical Linguistics - IATL 31*, pages 67–82, Ramata Gan, Israel, October.

Quochi, Valeria. 2007. *A usage-based approach to light verb constructions in Italian: Development and use*. Ph.D. thesis, University of Pisa, Pisa.

Ramisch, Carlos, Laurent Besacier, and Alexander Kobzar. 2013. How hard is it to automatically translate phrasal verbs from English to French. In *Proceedings of the MT Summit 2013 Workshop on Multi-word Units in Machine Translation and Translation Technology*, pages 53–61, Nice, France, September.

Ramisch, Carlos, Silvio Ricardo Cordeiro, Agata Savary, Veronika Vincze, Verginica Barbu Mititelu, Archna Bhatia, Maja Buljan, Marie Candito, Polona Gantar, Voula Giouli, Tunga Güngör, Abdelati Hawwari, Uxoa Iñurrieta, Jolanta Kovalevskaitė, Simon Krek, Timm Lichte, Chaya Liebeskind, Johanna Monti, Carla Parra Escartín, Behrang QasemiZadeh, Renata Ramisch, Nathan Schneider, Ivelina Stoyanova, Ashwini Vaidya, and Abigail Walsh. 2018. Edition 1.1 of the PARSEME Shared Task on Automatic Identification of Verbal

Multiword Expressions. In *Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018)*, Santa Fe, New Mexico, USA, August.
Ren, Zhixiang, Yajuan Lü, Jie Cao, Qun Liu, and Yun Huang. 2009. Improving statistical machine translation using domain bilingual multiword expressions. In *Proceedings of the Workshop on Multiword Expressions: Identification, Interpretation, Disambiguation and Applications*, pages 47–54, Singapore, August. Association for Computational Linguistics.
Reuland, Eric. 1990. Reflexives and Beyond: Non-local Anaphora in Italian Revisited. *Grammar in progress: glow essays for Henk van Riemsdijk*, 36:351.
Rikters, Matīss and Ondřej Bojar. 2017. Paying attention to multi-word expressions in neural machine translation. *arXiv preprint arXiv:1710.06313*.
Sag, Ivan A, Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. 2002. Multiword expressions: A pain in the neck for NLP. In *Proceedings of the International Conference on Intelligent Text Processing and Computational Linguistics*, pages 1–15, Mexico City, Mexico, February.
Salehi, Bahar, Paul Cook, and Timothy Baldwin. 2016. Determining the Multiword Expression Inventory of a Surprise Language. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 471–481, Osaka, Japan, December.
Savary, Agata and Silvio Cordeiro. 2018. Literal readings of multiword expressions: as scarce as hen's teeth. In *Proceedings of the 16th International Workshop on Treebanks and Linguistic Theories (TLT 16)*, pages 64–72, Prague, Czech Republic, January.
Savary, Agata, Carlos Ramisch, Silvio Cordeiro, Federico Sangati, Veronika Vincze, Behrang Qasemizadeh, Marie Candito, Fabienne Cap, Voula Giouli, Ivelina Stoyanova, et al. 2017. The PARSEME Shared Task on Automatic Identification of Verbal Multiword Expressions. In *Proceedings of the 13th Workshop on Multiword Expressions (MWE 2017)*, pages 31–47, Valencia, Spain, April.
Savary, Agata, Carlos Ramisch, Jena D. Hwang, Nathan Schneider, Melanie Andresen, Sameer Pradhan, and Miriam R. L. Petruck, editors. 2018. *Proceedings of the Joint Workshop on Linguistic Annotation, MultiWord Expressions and Constructions (LAW-MWE-CxG-2018)*, Santa Fe, New Mexico, USA, August. Association for Computational Linguistics.
Savary, Agata, Manfred Sailer, Yannick Parmentier, Michael Rosner, Victoria Rosén, Adam Przepiórkowski, Cvetana Krstev, Veronika Vincze, Beata Wójtowicz, Gyri Smørdal Losnegaard, et al. 2015. Parseme–parsing and multiword expressions within a European multilingual network. In *Proceedings of the 7th Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics (LTC 2015)*, Poznań, Poland, November.
Shapiro, Naomi Tachikawa. 2016. Splitting compounds with ngrams. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 630–640, Osaka, Japan, December.
Sheinfux, Livnat Herzig, Tali Arad Greshler, Nurit Melnik, and Shuly Wintner. 2019. Verbal multiword expressions: Idiomaticity and flexibility. In Yannick Parmentier and Jakub Waszczuk, editors, *Representation and parsing of multiword expressions*. Language Science Press, Berlin, Germany, pages 35–68.
Simone, Raffaele. 1997. Esistono verbi sintagmatici in italiano? In Tullio De Mauro, editor, *Lessico e grammatica. Teorie linguistiche e applicazioni lessicografiche - Atti del Convegno interannuale della Società di linguistica italiana*, pages 155–170, Madrid.
Stymne, Sara, Nicola Cancedda, and Lars Ahrenberg. 2013. Generation of compound words in statistical machine translation into compounding languages. *Computational Linguistics*, 39(4):1067–1108.
Tabossi, Patrizia, Lisa Arduino, and Rachele Fanari. 2011. Descriptive norms for 245 Italian idiomatic expressions. *Behavior Research Methods*, 43(1):110–123.
Taslimipoor, Shiva, Anna de Santis, Johanna Monti, et al. 2016. Language resources for Italian: towards the development of a corpus of annotated Italian multiword expressions. In *Proceedings of the Third Italian Conference on Computational Linguistics (CLiC-it 2016)*, pages 285–290, Napoli, Italy, December. Accademia University Press.
Venkatapathy, Sriram and Aravind Joshi. 2006. Using Information about Multi-Word Expressions for the Word-Alignment Task. In *Proceedings of the Workshop on Multiword Expressions: Identifying and Exploiting Underlying Properties*, pages 20–27, Sydney, Australia, July.

Vietri, Simonetta. 2014a. *Idiomatic Constructions in Italian: A Lexicon-grammar Approach*, volume 31. John Benjamins Publishing Company, Amsterdam/New York.
Vietri, Simonetta. 2014b. The Italian module for Nooj. In *Proceedings of the First Italian Conference on Computational Linguistics, CLiC-it*, pages 389–393, Pisa, Italy, December.
Vietri, Simonetta. 2014c. The Lexicon-Grammar of Italian idioms. In *Workshop on Lexical and Grammatical Resources for Language Processing, COLING 2014*, pages 137–146, Dublin, Ireland, August.
Villavicencio, Aline, Timothy Baldwin, and Benjamin Waldron. 2004. A Multilingual Database of Idioms. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal, May.
Viviani, Andrea. 2006. *I verbi procomplementari tra grammatica e lessicografia*. Le Lettere, Firenze, Italy.
Wehrli, Eric and Aline Villavicencio. 2015. Extraction of Multilingual MWEs from Aligned Corpora. In *PARSEME 5th General Meeting*, Iași, Romania, September.
Zaninello, Andrea and Alexandra Birch. 2020. Multiword expression aware neural machine translation. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 3816–3825, Marseille, France, May. European Language Resources Association.
Zaninello, Andrea and Malvina Nissim. 2010. Creation of lexical resources for a characterisation of multiword expressions in Italian. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, La Valletta, Malta, May.

In Memory of Emanuele Pianta's Contribution to Computational Linguistics

Bernardo Magnini*
Fondazione Bruno Kessler

Rodolfo Delmonte**
Università di Venezia

Sara Tonelli†
Fondazione Bruno Kessler

Almost eight years after his untimely death, the scientific contribution of Emanuele Pianta still appears significant to us, in particular for the variety of the topics he dealt with and for his capacity to move cross-disciplinarily between different areas of computational linguistics. Today, retracing the steps of Emanuele's scientific carrier has the meaning of rediscovering an important part of the scientific challenges that the Italian research community has faced over a period of more than twenty years. In recognition of the role he played, the Italian Association of Computational Linguistics entitled to Emanuele Pianta the annual award assigned to the best master's degree thesis in the context of Computational Linguistics, discussed in an Italian University.

1. Il percorso scientifico di Emanuele Pianta

Emanuele Pianta, scomparso nel novembre 2012 a causa di un incidente stradale, è stato uno dei ricercatori che maggiormente hanno contribuito alla crescita della Linguistica Computazionale in Italia, muovendosi con grande competenza in vari settori, dalla semantica lessicale, allo sviluppo di risorse linguistiche, all'analisi sintattica della frase, all'estrazione di informazioni da testi, in particolare entità nominate e concetti-chiave, agli algoritmi di semplificazione del testo, all'interpretazione semantica della frase, e infine aprendo nuove strade nel settore delle Digital Humanities.

Durante gli anni dell'Università al Dipartimento di Linguistica della Facoltà di Lettere e Filosofia di Padova, Emanuele matura i suoi interessi per la Linguistica Computazionale, in particolare per la generazione in linguaggio naturale. Dopo essersi laureato nel 1990 con una tesi su "Rilevanza e Rappresentazione - Preliminari Teorici a un Sistema per la Generazione Automatica del Linguaggio Naturale" presso l'Università di Padova (relatori i Professori Rodolfo Delmonte e Gianluigi Borgato), Emanuele ha collaborato con il laboratorio di Linguistica Computazionale alla Ca' Foscari di Venezia, diretto da Rodolfo Delmonte, e con l'azienda ICON di Verona, per poi passare all'Irst di Trento nel 1994, chiamato da Oliviero Stock.

L'attività scientifica di Emanuele Pianta ha attraversato circa due decenni, nel corso dei quali si è confrontato e ha dato un contributo importante allo sviluppo della

* Fondazione Bruno Kessler - Via Sommarive, 18, 38123 Povo TN, Italy. E-mail: magnini@fbk.eu
** Department of Linguistic Studies, Ca' Bembo - Dorsoduro 1075 30123 Venezia, Italy. E-mail: delmont@unive.it
† Fondazione Bruno Kessler - Via Sommarive, 18, 38123 Povo TN, Italy. E-mail: tonelli@fbk.eu

Linguistica Computazionale in Italia, in particolare sui temi delle risorse linguistiche, dell'analisi morfo-sintattica della frase, dell'estrazione di informazione da testo, della generazione a partire da contenuti strutturati, e infine, dei metodi di valutazione delle tecnologie del linguaggio. Visti i suoi numerosi interessi, durante i suoi anni di attività Emanuele ha stabilito numerose relazioni con altri ricercatori e gruppi di ricerca, sia in Italia che all'estero, ponendo le basi per progetti di ricerca che durano ancora nel tempo. Vogliamo sottolineare come Emanuele abbia sempre avuto una attitudine interdisciplinare alla ricerca, portandolo a coniugare la sua formazione linguistica con una forte attenzione ai metodi computazionali, in particolare quelli basati sull'apprendimento da dati linguistici, e anche con una non comune capacità di tradurre la ricerca in tecnologia e applicazioni.

Quando Emanuele è prematuramente scomparso l'Associazione Italiana di Linguistica Computazionale (AILC) ancora non esisteva, essendo stata fondata nel settembre del 2015. Ci sembra oggi che la sua attenzione alla ricerca multidisciplinare rappresenti bene lo spirito di AILC, nata con la missione di includere sotto un'unica iniziativa le diverse anime della Linguistica Computazionale in Italia. Intitolando a lui il premio per la miglior tesi magistrale, AILC riconosce a Emanuele Pianta l'importante ruolo svolto nell'avviare tematiche di ricerca basate sia sullo studio linguistico dei fenomeni sia sulla loro modellazione computazionale, realizzando soluzioni ancora oggi apprezzate.

Di seguito menzioniamo i principali progetti di ricerca in cui Emanuele è stato coinvolto.

Semantica Situazionale. Uno dei primi interessi di Emanuele è stata l'interpretazione semantica tramite linguaggi logici per la rappresentazione della frase in forma simbolica. Nel periodo a Venezia Emanuele ha lavorato al componente di Semantica Situazionale del sistema di analisi della lingua italiana GETARUNS (Delmonte, Bianchi, and Pianta 1992), contribuendo alla implementazione del modulo che trasferisce il contenuto del DAG (grafo diretto aciclico) con l'informazione sintattica alla Forma Logica, dopo aver elaborato la risoluzione delle referenze pronominali.

MultiWordNet. Insieme ad alcuni colleghi dell'IRST di Trento (ora Fondazione Bruno Kessler) Emanuele ha contribuito alla progettazione e alla realizzazione di MultiWordNet, la versione italiana di WordNet, fin dal suo inizio (Magnini et al. 1994b), (Magnini et al. 1994a), nel 1994. Negli anni successivi Emanuele divenne il riferimento per una serie di attività di ricerca legate alle metodologie di sviluppo di wordnet multilingui allineati, includendo studi sui "lexical gap" e sulla possibilità di trasferire annotazioni semantiche da una lingua ad altre (e.g. MultiSemcor (Bentivogli and Pianta 2005)). La metodologia sperimentata con MultiWordNet è stata adottata per diverse lingue, inclusa una originale versione per il latino, e la risorsa è stata distribuita in diverse migliaia di licenze d'uso.

Traduzione speech to speech. Alla fine degli anni '90 Emanuele ha svolto un ruolo importante all'interno del progetto NESPOLE, portando le proprie competenze di linguistica computazionale in un contesto di collaborazioni internazionali sul tema della traduzione automatica speech-to-speech. Uno dei risultati di rilievo è stato un dataset multilingua (Mana et al. 2004), che raccoglie dialoghi parlati nei domini del turismo e della medicina, con le loro trascrizioni e annotazioni a livello di interlingua.

CELCT. Per il periodo da giugno 2009 a novembre 2012 Emanuele ha assunto la direzione scientifica di CELCT, il "Centro per la valutazione delle tecnologie del linguag-

gio e della comunicazione" di Trento, subentrando a Amedeo Cappelli, che ne era stato direttore dal 2003. Fondamentali sono stati i contributi di Emanuele per lo sviluppo di una serie di benchmark per la lingua italiana, tra cui I-CAB (Magnini et al. 2006), ancora oggi utilizzato come data set di addestramento per task di estrazione di informazione da testi, e la versione italiana di Time-ML (Caselli et al. 2011).

Evalita. Sotto la direzione di Emanuele, CELCT ha contribuito in particolare all'organizzazione di Evalita 2011, la campagna di valutazione delle tecnologie del linguaggio scritto e parlato, per la lingua italiana, di cui fu co-coordinatore scientifico (Magnini et al. 2012). Si devono in buona parte al contributo di CELCT i task su Named Entity Recognition on Transcribed Broadcast News e Cross-document Coreference Resolution of Named Person Entities in quella edizione di Evalita. In quanto direttore del Centro, Emanuele fu anche responsabile dei numerosi progetti che hanno coinvolto CELCT, e che hanno fatto di Emanuele una figura molto conosciuta e apprezzata anche a livello internazionale.

TextPro. Uno dei maggiori risultati tecnologici raggiunto da Emanuele è stata l'ideazione e la realizzazione della piattaforma TextPro (Pianta, Girardi, and Zanoli 2008) per l'annotazione di informazione su testi. TextPro è stato progettato come una cascata di annotatori indipendenti (tokenizzatore, post tagging, analizzatore morfo-sintattico, riconoscitore di entità nominate, ecc.) raggruppati in una unica piattaforma. Progettata inizialmente per l'italiano, TextPro è stato in seguito esteso all'inglese, e il piano iniziale arricchito con ulteriori moduli di annotazione. La gran parte dei progetti applicativi nel campo delle tecnologie del linguaggio portati avanti da FBK, per anni si è avvalsa di TextPro come strumento di estrazione di informazioni da testi scritti.

FrameNet. Dopo MultiWordNet, Emanuele si è dedicato alla creazione di FrameNet per l'italiano (Tonelli and Pianta 2008), una risorsa semantica per categorizzare situazioni e eventi in "frame", e i relativi partecipanti in "frame element", o ruoli semantici. Partendo da FrameNet per l'inglese, sviluppato alla fine degli anni '90 a Berkeley sulla base della "frame semantics" proposta dal linguista Charles Fillmore, Emanuele ha proposto di crearne la versione italiana, riutilizzando dove possibile tecniche di proiezione dell'annotazione già sperimentate in MultiWordNet. La risorsa annotata, da lui coordinata, è stata rilasciata alla comunità scientifica e rappresenta tutt'ora uno dei nuclei centrali di FrameNet per l'italiano (Basili et al. 2017), un progetto ancora in corso a cui collaborano diverse università.

Lessico bilingue della lingua veneta. Per un breve periodo Emanuele ha collaborato al progetto STILVEN sulla lingua veneta, finanziato dalla Regione Veneto, producendo un lessico bilingue con le forme di parola morfologiche di tutti i verbi - solo lemmi - inclusi nei dizionari già disponibili, incluse le forme cliticizzate.

Parole chiave. Gli interessi scientifici di Emanuele nascevano spesso da esigenze pratiche. Per esempio, l'idea di implementare un estrattore di concetti-chiave multilingua era stato pensato come un primo passo per arrivare alla generazione automatica di mappe concettuali, che gli studenti potessero utilizzare a scopi educativi. Anche se il tema delle mappe concettuali è rimasto purtroppo inesplorato, Emanuele ha ideato, implementato e rilasciato il sistema Keyword eXtractor (KX) (Pianta and Tonelli 2010), un estrattore di concetti-chiave configurabile a seconda del dominio, basato su criteri linguistici per il riconoscimento di espressioni polirematiche. Il tool ha dimostrato la propria efficacia in

ambiti diversi, dall'analisi di testi brevettuali (progetto Patexpert) a quella di documenti storici (progetto Alcide).

Semplificazione del testo. Un altro ambito di studio a cui Emanuele si è dedicato è stato quello della profilazione del testo finalizzata a comprendere quali aspetti di un documento potevano risultare di difficile comprensione, soprattutto per bambini con disabilità cognitive. Questo problema è stato affrontato da Emanuele con un approccio interdisciplinare che coniugava l'analisi e la generazione di linguaggio naturale con le scienze cognitive, il design di interfacce uomo-macchina e la gamification. Le tecnologie sviluppate da Emanuele nei progetti LODE e Terence sono state utilizzate con successo da bambini non udenti e da quelli con lievi disabilità cognitive, che hanno potuto giocare e fare esercizi a partire da storie semplificate con metodi automatici.

2. Il premio AILC "Emanuele Pianta" per la miglior tesi di laurea magistrale

Alla luce dei suoi numerosi contributi scientifici, il Consiglio Direttivo dell'Associazione Italiana di Linguistica Computazionale, nella seduta del 12 febbraio 2020, ha deciso all'unanimità di intitolare a Emanuele Pianta il premio annuale assegnato alla miglior tesi di laurea magistrale nell'ambito della Linguistica Computazionale, discussa in una università italiana.

Il premio AILC è stato istituito nel 2017 in corrispondenza della quarta edizione della conferenza CLiC-it, svoltasi a Roma dall'11 al 13 dicembre 2017, con l'obiettivo di promuovere e individuare eccellenze nel campo della ricerca della Linguistica Computazionale (vengono considerate le aree elencate nella call for papers della conferenza CLiC-it). Il premio viene assegnato da una giuria composta da tre membri: un membro del comitato organizzatore del convegno CLiC-it dell'anno precedente, un membro del comitato organizzatore del convegno CLiC-it dell'anno in corso (questo membro si impegna a servire nella giuria per due anni, così da garantire continuità) e un membro del Direttivo AILC. Il premio consiste in 500 euro, l'iscrizione gratuita a AILC per un anno, e l'iscrizione alla conferenza CLiC-it, dove l'autore ha la possibilità di presentare la tesi vincitrice del premio.

Giunto alla terza edizione, il premio si è affermato nella comunità di ricerca italiana come un riconoscimento importante a studenti brillanti nel settore della Linguistica Computazionale. In ordine temporale, il premio è stato assegnato a Alessio Miaschi (2017 - Università di Pisa, "Definizione di modelli computazionali per lo studio dell'evoluzione delle abilità di scrittura a partire da un corpus di produzioni scritte di apprendenti della scuola secondaria di primo grado"), Enrica Troiano (2018 - Università di Trento/FBK, "A Computational Study of Linguistic Exaggerations") e Ludovica Pannitto (2019 - Università di Pisa, "Event Knowledge in Compositional Distributional Semantics").

Ci piace concludere questo breve ricordo della figura di Emanuele Pianta riassumento gli aspetti che, a nostro parere, hanno maggiormente caratterizzato il suo contributo nel campo della Linguisitica Computazionale. In primo luogo l'attitudine alla ricerca multidisciplinare, con lo scopo di combinare le conoscenze acquisite in ambiti diversi, nella convinzione che questa combinazione possa portare ad una migliore comprensione della complessità sottostante all'uso del linguaggio. Poi la visione sulle direzioni della ricerca, ad esempio intuendo l'importanza di puntare sulla piattaforma TextPro, oppure sullo sviluppo di FrameNet per l'italiano. Infine, l'impatto di Emanuele nel nostro campo è stato possibile anche grazie alla sua innata capacità di comunicare,

con la quale ha coinvolto tutti, giovani studenti e ricercatori ormai affermati, in appassionate discussioni sulla Linguisitica Computazionale.

Tutto questo ha motivato AILC nella scelta di intitolare ad Emanuelel Pianta il premio per la miglior tesi magistrale in Linguistca Computazionale, e rende Emanuele un esempio per le generazioni future.

References

Basili, Roberto, Silvia Brambilla, Danilo Croce, and Fabio Tamburini. 2017. Developing a large scale FrameNet for Italian: the IFrameNet experience. In *Proceedings of the Fourth Italian Conference on Computational Linguistics (CLiC-it 2017)*, Rome, 11-13 December.

Bentivogli, Luisa and Emanuele Pianta. 2005. Exploiting parallel texts in the creation of multilingual semantically annotated resources: The MultiSemCor Corpus. *Natural Language Engineering, Special Issue on Parallel Texts*, 11(3):247–261.

Caselli, Tommaso, Valentina Bartalesi Lenzi, Rachele Sprugnoli, Emanuele Pianta, and Irina Prodanof. 2011. Annotating events, temporal expressions and relations in Italian: the It-TimeML experience for the Ita-TimeBank. In *Proceedings of the 5th Linguistic Annotation Workshop*, pages 143–151, Portland, Oregon, USA, June 23-24.

Delmonte, Rodolfo, Dario Bianchi, and Emanuele Pianta. 1992. GETA_RUN - A general text analyzer with reference understanding. In *Proceedings of the 3rd Conference on Applied Natural Language Processing, Systems Demonstrations*, pages 9–10, Trento, Italy, March.

Magnini, Bernardo, Francesco Cutugno, Mauro Falcone, and Emanuele Pianta, editors. 2012. *Evaluation of Natural Language and Speech Tools for Italian, International Workshop, EVALITA 2011, Rome, January 24-25, 2012, Revised Selected Papers*. Springer.

Magnini, Bernardo, Emanuele Pianta, Christian Girardi, Matteo Negri, Lorenza Romano, Manuela Speranza, Valentina Bartalesi Lenzi, and Rachele Sprugnoli. 2006. I-CAB: the Italian Content Annotation Bank. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC 2006)*, Genoa, Italy, May.

Magnini, Bernardo, Carlo Strapparava, Fabio Ciravegna, and Emanuele Pianta. 1994a. Multilingual lexical knowledge bases: Applied WordNet prospects. In *Proceedings of the International Workshop on the Future of the Dictionary*, Grenoble, October.

Magnini, Bernardo, Carlo Strapparava, Fabio Ciravegna, and Emanuele Pianta. 1994b. A project for the construction of an Italian lexical knowledge base in the framework of WordNet. Technical report, IRST # 9406-15, June.

Mana, Nadia, Roldano Cattoni, Emanuele Pianta, Franca Rossi, Fabio Pianesi, and Susanne Burger. 2004. The Italian NESPOLE! corpus: a multilingual database with interlingua annotation in tourism and medical domains. In *Proceedings of the fourth International Conference on Language Resources and Evaluation (LREC-2004)*, Lisbon, Portugal, May.

Pianta, Emanuele, Christian Girardi, and Roberto Zanoli. 2008. The TextPro tool suite. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008)*, Marrakech, Morocco, May.

Pianta, Emanuele and Sara Tonelli. 2010. KX: A flexible system for keyphrase extraction. In *Proceedings of the 5th International Workshop on Semantic Evaluation (SemEval '10) at ACL 2010*, pages 170—173, Uppsala, Sweden, July.

Tonelli, Sara and Emanuele Pianta. 2008. Frame information transfer from English to Italian. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008)*, Marrakech, Morocco, May.