IJCoL

Italian Journal of Computational Linguistics

Rivista Italiana di Linguistica Computazionale

Volume 4, Number 1
june 2018

Emerging Topics at the Fourth Italian Conference on Computational Linguistics

aAccademia university press

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

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

Associazione Italiana di Linguistica Computazionale

direttore responsabile
Michele Arnese

isbn 978-88-31978-40-8

Accademia University Press
via Carlo Alberto 55
I-10123 Torino
info@aAccademia.it
www.aAccademia.it/IJCoL_4_1

IJCoL

Volume 4, Number 1
june 2018

CONTENTS

Emerging Topics at the Fourth Italian Conference on Computational Linguistics

Roberto Basili*
Università di Roma, Tor Vergata

Simonetta Montemagni**
ILC - CNR

1. Introduction

E' con gran piacere che introduciamo il primo volume del quarto anno della rivista *Italian Journal of Computational Linguistics* (*IJCoL*), la rivista italiana di linguistica computazionale promossa dall'*Associazione Italiana di Linguistica Computazionale* (AILC - www.ai-lc.it). La rivista, fino a oggi, è uscita regolarmente con cadenza semestrale e ha raccolto importanti contributi della comunità nazionale e internazionale della linguistica computazionale, con particolare attenzione a ricerca di frontiera condotta da parte di giovani ricercatori. I numeri pubblicati finora coprono un ampio spettro di temi che ruotano attorno alla dicotomia linguaggio-computazione, affrontata da prospettive diverse riconducibili alle "anime" umanistica e informatica della linguistica computazionale, con diverse finalità, sia teoriche sia applicative, e con particolare attenzione al trattamento automatico della lingua italiana nelle sue diverse varietà d'uso. Dalla fondazione, sono stati pubblicati due numeri speciali della rivista, dedicati all'approfondimento di aree di ricerca strategiche della disciplina, sul versante umanistico e informatico, riguardanti rispettivamente l'apporto di metodi e tecniche della linguistica computazionale alle "Digital Humanities" e i paradigmi dominanti nel panorama degli algoritmi di apprendimento automatico che stanno influenzando pesantemente gli sviluppi correnti della disciplina.

Dalla fase iniziale di rodaggio la rivista sta passando oggi a una fase più matura: ne è testimonianza il recente riconoscimento da parte dell'*Academic Committee* di OpenEdition, l'infrastruttura europea dedicata alla comunicazione e alla pubblicazione in Open Access della ricerca accademica in un ampio spettro di settori scientifici, che ha deliberato la pubblicazione della rivista sulla piattaforma *OpenEdition Journals*. Siamo consapevoli che per una nuova rivista la strada per guadagnare prestigio e autorevolezza è lunga e tutt'altro che scontata. Crediamo tuttavia che i risultati conseguiti finora siano promettenti e stiano creando i presupposti per la classificazione di IJCoL tra le riviste scientifiche del settore e, in prospettiva, tra le riviste in Classe A dell'ANVUR, e per la sua indicizzazione nei principali database internazionali rilevanti per i settori coperti dalla rivista (tra i quali, Scopus Bibliographic Database, ERIH Plus, Google Scholar, Web of Science). Tutto ciò, grazie alla passione e all'impegno di chi, a diverso titolo, sta contribuendo a questa importante impresa.

Alla Special Issue su *Natural Language and Learning Machines* (n. 3, vol. 2, 2017), segue oggi questo numero miscellaneo che, seguendo la tradizione di diversi precedenti

* 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

numeri, raccoglie lavori di ricerca ispirati da giovani ricercatori, che sono emersi come particolarmente promettenti nell'ambito della Conferenza CLiC-it 2017, tenutasi a Roma dall'11 al 13 dicembre 2017. Questo insieme corrisponde a una prima selezione di contributi di CLiC-it 2017, caratterizzata dalla ricerca su algoritmi di apprendimento profondo ("deep learning") per la soluzione di diversi e complessi compiti di inferenza linguistica. Si presenta quindi come una sorta di continuazione del precedente numero speciale della rivista, preludendo così a una seconda selezione dei lavori da CLiC-it 2017, la cui pubblicazione è prevista per il prossimo numero del 2018.

Come per altri numeri miscellanei della rivista, gli articoli di questo numero sono stati selezionati attraverso un processo iterativo di *peer-review*. Ogni articolo è stato sottoposto a tre valutazioni da parte di comitati diversi: come contributo alla conferenza; come candidato ai premi di "Best Young Paper" e "Distinguished Young Paper" di CLiC-it 2017; infine, nella versione estesa, come articolo di rivista scientifica. A questi articoli si aggiunge il contributo invitato dedicato alla rassegna della Conferenza CLiC-it 2017, a cura dei tre *co-chair*, con particolare attenzione alle novità introdotte per un coinvolgimento sempre maggiore della comunità italiana della linguistica computazionale: dei giovani all'interno di percorsi di formazione così come dei potenziali "stakeholders" - che vanno dalla Pubblica Amministrazione alle piccole e medie imprese - come beneficiari dell'apporto dei risultati della ricerca nazionale e internazionale nel settore della linguistica computazionale.

Apre il volume il paper di Lakew e colleghi, che discute un modello neurale per la traduzione automatica in grado di affrontare la sfida posta da lingue caratterizzate da una scarsa disponibilità di risorse di *training*. L'approccio proposto si basa sulla creazione di uno spazio semantico multilingue che permette il trasferimento dei parametri usati tra lingue diverse, migliorando così le condizioni di addestramento per i casi in cui i dati per una specifica coppia di lingue siano limitati. Nel lavoro, questa ipotesi è verificata mostrando i risultati di esperimenti condotti su tre lingue (inglese, italiano e rumeno): la metodologia proposta migliora le prestazioni rispetto a sistemi bilingui, evitando al contempo la complessità insita nell'addestramento di tali sistemi.

Nel lavoro di Bizzoni et al., rappresentazioni vettoriali sono utilizzate per la classificazione di espressioni idiomatiche e non, in condizioni di limitata disponibilità di dati: l'obiettivo è quello di verificare se l'informazione convogliata dal vettore distribuzionale associato a una data espressione sia sufficiente alla rete per inferire la sua potenziale idiomaticità. La sperimentazione presentata conferma il ruolo cruciale di rappresentazioni distribuzionali in questo compito. Diversamente da quanto rilevato in precedenza per il riconoscimento di espressioni metaforiche, l'impiego di rappresentazioni vettoriali associate all'espressione nel suo complesso si dimostra essere più efficace rispetto alla concatenazione dei vettori associati alle singole parole dell'espressione.

Il lavoro di Masotti e colleghi, si occupa di un tema piuttosto nuovo nel panorama italiano: la generazione automatica di didascalie per immagini, processo che coinvolge in modo integrato competenze di tipo visuale (nel riconoscimento dei tipi di oggetti ritratti nella immagine) e linguistico (nella generazione di frasi corrette che descrivono gli oggetti e la situazione ritratta che li coinvolge). L'architettura neurale presentata integra due reti distinte: una prima rete dedicata all'*embedding* grafico, e una seconda rete ricorrente per la generazione automatica della didascalia che utilizza l'*embedding* prodotto dalla prima come input (stato iniziale). Tale architettura, già applicata con successo alla lingua inglese, è stata addestrata su un dataset esteso per la lingua italiana ottenuto attraverso strumenti di traduzione automatica applicati alle descrizioni in inglese di una collezione di immagini.

Infine, il lavoro di Cardillo e colleghi affronta il problema dell'induzione di conoscenza morfologica seguendo un approccio che, piuttosto che presupporre una segmentazione delle parole in morfemi, si basa sulle connessioni tra forme flesse all'interno di reti lessicali associative. Secondo questo approccio, l'identificazione della cella paradigmatica appropriata per una forma flessa sconosciuta è guidata dall'evidenza offerta da forme conosciute. La novità del contributo consiste nell'utilizzo di reti neurali per modellare il processo di flessione delle parole come inferenza paradigmatica. A tal fine, sono state utilizzate reti di tipo *Long Short Term Memory* (LSTM) che si sono mostrate particolarmente flessibili ed efficaci nel combinare diversi tipi di informazione (relativa alla struttura morfologica, all'organizzazione paradigmatica e al grado di (ir)regolarità nella formazione del tema), e in grado di adattarsi alle specificità e ai diversi livelli di complessità caratterizzanti ciascun sistema morfologico.

Questa breve vista d'insieme non esaurisce i molti aspetti di interesse che emergono dai lavori che compongono il presente volume per quanto concerne l'adozione di tecnologie di apprendimento automatico per il trattamento della lingua. Lasciamo quindi al lettore l'onere e il piacere di approfondirli direttamente negli articoli qui raccolti.

2. Editorial Note Summary

It is with great pleasure that we introduce the first volume of the fourth year of the *Italian Journal of Computational Linguistics* (IJCoL) promoted by the *Associazione Italiana di Linguistica Computazionale* (AILC - `www.ai-lc.it`). Until today, the journal has been regularly published biannually and has collected important contributions from the national and international communities of computational linguistics, with particular attention to frontier research carried out by young researchers. The volumes published so far cover a wide spectrum of themes revolving around the language-computation dichotomy, addressed from the humanistic and computational perspectives, with both theoretical and applicative purposes, and with particular emphasis on the automatic processing of Italian in its different varieties of use.

Since its foundation, two special issues have been published, dedicated to strategic areas of the discipline: namely, the contribution of methods and techniques of computational linguistics to "Digital Humanities" and the dominant Machine Learning paradigms that are currently influencing the developments of the discipline. The journal is now entering into a mature phase, as confirmed e.g. by the recent recognition by the *OpenEdition Academic Committee* which has deliberated the publication of IJCoL on the *OpenEdition Journals* platform. We are aware that it takes time to earn prestige for a new journal. However, we believe that the results achieved so far are promising and are creating the prerequisites for the classification of IJCoL among the scientific journals in the computational linguistics area and for its indexing in the main international bibliographic databases. All this has been possible thanks to the passion and commitment of those who, in different ways, are contributing to this important enterprise.

This miscellaneous volume follows the Special Issue on *Natural Language and Learning Machines* (No. 3, Volume 2, 2017); as in the previous issues, it collects a selection of research contributions inspired by young researchers which emerged as particularly promising at the CLiC-it 2017 Conference, held in Rome from 11 to 13 December 2017. This first selection of contributions from CLiC-it 2017 shares the use of deep learning algorithms for the solution of different and challenging linguistic problems. It thus presents itself as a sort of continuation of the previous special issue of the journal, which will be followed by another miscellaneous volume with a second selection of papers from CLiC-it 2017, whose publication is scheduled for the second issue of 2018.

As for the other miscellaneous issues, the papers have been selected through an iterative peer-review process. Each article underwent three evaluations: as a contribution to the conference; as a candidate for the "Best Young Paper" and "Distinguished Young Paper" awards of CLiC-it 2017; finally, in the extended version, as a journal article. This set of papers also includes an invited contribution devoted to a retrospective of CLiC-it 2017 by the conference co-chair, with particular attention to the innovations introduced for an increasing involvement of the Italian community of computational linguistics: in particular, young researchers and potential "stakeholders", ranging from public administrations to small and medium-sized companies.

The volume opens with the paper by Lakew and colleagues, discussing a neural model for Machine Translation (NMT) that addresses the challenge of low-resourced languages. The proposed approach is multilingual: i.e. it is based on the creation of hidden representations of words in a shared semantic space across multiple languages, thus enabling a positive parameter transfer across languages. Results of experiments carried out on three languages (English, Italian and Romanian) are reported: compared to bilingual NMT systems, the system significantly improves its performance, while avoiding the complexity inherent in training systems for single language pairs.

In the paper by Bizzoni et al. vector representations are used for the classification of Italian idiomatic and non-idiomatic phrases under constrains of data scarcity. The goal is to assess whether and to what extent the information conveyed by the distributional vector associated with a phrase (whether idiomatic or not) is sufficient for the network to infer its potential idiomaticity. Reported experiments confirm the crucial role of distributional representations in this task. Contrary to what previously reported for metaphorical expressions, the use of phrase-based vector representations proves to be more effective than the concatenation of the vectors associated with the individual words of the expression.

The work by Masotti and colleagues tackles a recent topic in the Italian landscape: the automatic generation of image captions, a process that involves both visual and linguistic skills. The neural architecture presented for this purpose integrates two distinct networks: a first network dedicated to the vector representation of the image, and a second recurrent network for the automatic generation of the caption that uses the *embedding* produced by the first as input. This architecture, already successfully applied to English, is trained on an extended data set for Italian obtained through automatic translation of English descriptions of a collection of images.

Finally, the work by Cardillo et al. addresses the problem of the induction of morphological knowledge following an approach that, rather than presupposing a segmentation of words into morphemes, is based on the connections between inflected forms within associative lexical networks. According to this approach, the identification of the appropriate paradigmatic cell for an unknown inflected form is guided by the evidence offered by known forms. The novelty of the contribution consists in the use of neural networks to model word inflection as a paradigmatic inference. To this end, *Long Short Term Memory* (LSTM) networks were used which proved to be particularly flexible and effective in combining different types of information and able to adapt to the peculiarities of each morphological system.

This synthetic view does not exhaust the wide range of issues touched by the papers and this leaves the reader the pleasure to discover them through a thoughtful sailing across the rest of the volume contents. We think this volume sheds further light on achievements regularly emerging from the worldwide dimensions of the computational linguistics research, with particular emphasis on the contributions by the Italian community.

Multilingual Neural Machine Translation for Low-Resource Languages

Surafel M. Lakew*
Fondazione Bruno Kessler
Universitá di Trento

Marcello Federico
MMT Srl, Trento
Fondazione Bruno Kessler

Matteo Negri
Fondazione Bruno Kessler

Marco Turchi
Fondazione Bruno Kessler

In recent years, Neural Machine Translation (NMT) has been shown to be more effective than phrase-based statistical methods, thus quickly becoming the state of the art in machine translation (MT). However, NMT systems are limited in translating low-resourced languages, due to the significant amount of parallel data that is required to learn useful mappings between languages. In this work, we show how the so-called multilingual NMT can help to tackle the challenges associated with low-resourced language translation. The underlying principle of multilingual NMT is to force the creation of hidden representations of words in a shared semantic space across multiple languages, thus enabling a positive parameter transfer across languages. Along this direction, we present multilingual translation experiments with three languages (English, Italian, Romanian) covering six translation directions, utilizing both recurrent neural networks and transformer (or self-attentive) neural networks. We then focus on the zero-shot translation problem, that is how to leverage multi-lingual data in order to learn translation directions that are not covered by the available training material. To this aim, we introduce our recently proposed iterative self-training method, which incrementally improves a multilingual NMT on a zero-shot direction by just relying on monolingual data. Our results on TED talks data show that multilingual NMT outperforms conventional bilingual NMT, that the transformer NMT outperforms recurrent NMT, and that zero-shot NMT outperforms conventional pivoting methods and even matches the performance of a fully-trained bilingual system.

1. Introduction

Neural machine translation (NMT) has shown its effectiveness by delivering the best performance in the IWSLT (Cettolo et al. 2016) and WMT (Bojar et al. 2016) evaluation campaigns of the last three years. Unlike rule-based or statistical MT, the end-to-end learning approach of NMT models the mapping from source to target language directly through a posterior probability. The essential component of an NMT system includes an encoder, a decoder and an attention mechanism (Bahdanau, Cho, and Bengio 2014). Despite the continuous improvement in performance and translation quality (Bentivogli et al. 2018), NMT models are highly dependent on the availability of extensive parallel data, which in practice can only be acquired for a very limited number of language pairs.

* Fondazione Bruno Kessler, Via Somarive 18, 38123 Povo (Trento), Italy. E-mail: lakew@fbk.eu

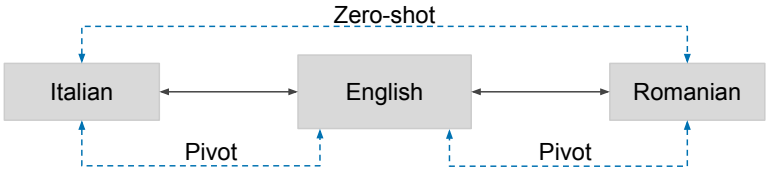

Figure 1
A multilingual setting with parallel training data in four translation directions: Italian↔English and Romanian↔English. Translations between Italian and Romanian are either inferred directly (zero-shot) or by translating through English (pivoting).

For this reason, building effective NMT systems for low-resource languages becomes a primary challenge (Koehn and Knowles 2017). In fact, Zoph et al. (2016) showed how a standard string-to-tree statistical MT system (Galley et al. 2006) can effectively outperform NMT methods for low-resource languages, such as Hausa, Uzbek, and Urdu.

In this work, we approach low-resource machine translation with so-called *multilingual* NMT (Johnson et al. 2017; Ha, Niehues, and Waibel 2016), which considers the use of NMT to target many-to-many translation directions. Our motivation is that intensive cross-lingual transfer (Odlin 1989) via parameter sharing across multiple languages should ideally help in the case of similar languages and sparse training data. In particular, we investigate multilingual NMT across Italian, Romanian, and English, and simulate low-resource conditions by limiting the amount of available parallel data.

Among the various approaches for multilingual NMT, the simplest and most effective one is to train a single neural network on parallel data including multiple translation directions and to prepend to each source sentence a *flag* specifying the desired target language (Ha, Niehues, and Waibel 2016; Johnson et al. 2017). In this work, we investigate multi-lingual NMT under low-resource conditions and with two popular NMT architectures: recurrent LSTM-based NMT and the self-attentive or transformer NMT model. In particular, we train and evaluate our systems on a collection of TED talks (Cettolo, Girardi, and Federico 2012), over six translation directions: English↔Italian, English↔Romanian, and Italian↔Romanian.

A major advantage of multi-lingual NMT is the possibility to perform a zero-shot translation, that is to query the system on a direction for which no training data was provided. The case we consider is illustrated in Figure 1: we assume to only have Italian-English and English-Romanian training data and that we need to translate between Italian and Romanian in both directions. To solve this task, we propose a *self-learning* method that permits a multi-lingual NMT system trained on the above mentioned language pairs to progressively learn how to translate between Romanian and Italian directly from its own translations. We show that our zero-shot self-training approach not only improves over a conventional pivoting approach, by bridging Romanian-Italian through English (see Figure 1), but that it also matches the performance of bilingual systems trained on Italian-Romanian data. The contribution of this work is twofold:[1]

- Comparing RNN and Transformer approaches in a multilingual NMT setting;

1 This paper integrates and extends work presented in (Lakew, Di Gangi, and Federico 2017) and (Lakew et al. 2017).

- A self-learning approach to improve the zero-shot translation task of a multilingual model.

The paper is organized as follows. In Section 2, we present previous works on multilingual NMT, zero-shot NMT, and NMT training with self-generated data. In Section 3, we introduce the two prominent NMT approaches evaluated in this paper, the recurrent and the transformer models. In Section 4, we introduce our multilingual NMT approach and our self-training method for zero-shot learning. In Section 5, we describe our experimental set-up and the NMT model configurations. In Section 6, we present and discuss the results of our experiments. Section 7 ends the paper with our conclusions.

2. Previous Work

2.1 Multilingual NMT

Previous works in multilingual NMT are characterized by the use of separate encoding and/or decoding networks for each translation direction. Dong et al. (2015) proposed a multi-task learning approach for a *one-to-many* translation scenario, by sharing hidden representations among related tasks – *i.e* the source languages – to enhance generalization on the target language. In particular, they used a single encoder for all source languages and separate attention mechanisms and decoders for every target language. In a related work, Luong et al. (2016) used distinct encoder and decoder networks for modeling language pairs in a *many-to-many* setting. Aimed at reducing ambiguities at translation time, Zoph and Knight (2016) employed a *many-to-one* system that considers two languages on the encoder side and one target language on the decoder side. In particular, the attention model is applied to a combination of the two encoder states. In a *many-to-many* translation scenario, Firat, Cho, and Bengio (2016) introduced a way to share the attention mechanism across multiple languages. As in Dong et al. (2015) (but only on the decoder side) and in Luong et al. (2016), they used separate encoders and decoders for each source and target language.

Despite the reported improvements, the need of using an additional encoder and/or decoder for every language added to the system tells the limitation of these approaches, by making their network complex and expensive to train.

In a very different way, Johnson et al. (2017) and Ha, Niehues, and Waibel (2016) developed similar multilingual NMT approaches by introducing a *target-forcing* token in the input. The approach in Ha, Niehues, and Waibel (2016) applies a language-specific code to words from different languages in a mixed-language vocabulary. In practice, they force the decoder to translate into a specific target language by prepending and appending an artificial token to the source text. However, their word and sub-word level language-specific coding mechanism significantly increases the input length, which shows to have an impact on the computational cost and performance of NMT (Cho et al. 2014a). In Johnson et al. (2017), only one artificial token is pre-pended to the entire source sentences in order to specify the target language. Hence, the same token is also used to trigger the decoder generating the translation (cf. Figure 2). Remarkably, prepending language tokens to the input string has greatly simplified multi-lingual NMT, by eliminating the need of having separate encoder/decoder networks and attention mechanism for every new language pair.

2.2 Zero-Shot and Self-Learning

Zero-resource NMT has been proposed in (Firat et al. 2016) and it extends the work by Firat, Cho, and Bengio (2016). The authors proposed a *many-to-one* translation setting and used the idea of generating a pseudo-parallel corpus (Sennrich, Haddow, and Birch 2015a), using a pivot language, to fine tune their model. However, also in this case the need of separate encoders and decoders for every language pair significantly increases the complexity of the model.

An attractive feature of the *target-forcing* mechanism comes from the possibility to perform zero-shot translation with the same multilingual setting as in (Johnson et al. 2017; Ha, Niehues, and Waibel 2016). Both the works reported that a multilingual system trained on a large amount of data improves over a baseline bilingual model and that it is also capable of performing zero-shot translation, assuming that the zero-shot source and target languages have been observed during training paired with some other languages.

However, recent experiments have shown that the mechanism fails to achieve reasonable zero-shot translation performance for low-resource languages (Lakew, Di Gangi, and Federico 2017). The promising results in (Johnson et al. 2017) and (Ha, Niehues, and Waibel 2016) hence require further investigation to verify if their method can work in various language settings, particularly across distant languages.

An alternative approach to zero-shot translation in a resource-scarce scenario is to use a pivot language (Cettolo, Bertoldi, and Federico 2011), that is, using an intermediate language for translation. While this solution is usually pursued by deploying two or more bilingual models, in this work we aim to achieve comparable results using a single multilingual model.

Training procedures using synthetic data have been around for a while. For instance, in statistical machine translation (SMT), Oflazer and El-Kahlout (2007) and Béchara, Ma, and van Genabith (2011) showed how the output of a translation model can be used iteratively to improve results in a task like post-editing. Mechanisms like back-translating the target side of a single language pair have been used for domain adaptation (Bertoldi and Federico 2009) and more recently by Sennrich, Haddow, and Birch (2015a) to improve an NMT baseline model. In (He et al. 2016), a dual-learning mechanism is proposed where two NMT models working in the opposite directions provide each other feedback signals that permit them to learn from monolingual data. In a related way, our approach also considers training from monolingual data. As a difference, however, our proposed method leverages the capability of the network to jointly learn multiple translation directions and to directly generate the translations used for self-training.

Although our brief survey shows that re-using the output of an MT system for further training and improvement has been successfully applied in different settings, our approach differs from past works in two aspects: *i)* introducing a new self-training method integrated in a multilingual NMT architecture, and *ii)* casting the approach into a *self-correcting* procedure over two dual zero-shot directions, so that incrementally improved translations mutually reinforce each direction.

3. Neural Machine Translation

State-of-the-art NMT systems comprise an encoder, a decoder, and an attention mechanism, which are jointly trained with maximum likelihood in an end-to-end fashion (Bahdanau, Cho, and Bengio 2014). Among the different variants, two popular ones are the recurrent NMT (Sutskever, Vinyals, and Le 2014) and the transformer NMT (Vaswani

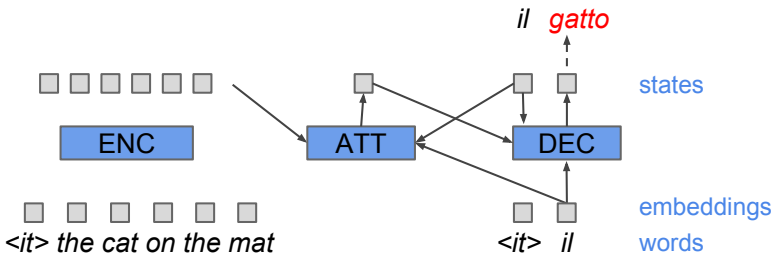

Figure 2
Encoder-decoder-attention NMT architecture. Once the encoder has generated his states for all the input words, the decoder starts generating the translation word by word. The target word "gatto" (cat) is generated from the previously generated target word "il" (the), the previous decoder state and the context state. The context state is a selection and combination of encoder states computed by the attention model. Finally, notice that the target-language forcing symbol "<it>" (Italian), prepended to the input, is also used to trigger the first output word.

et al. 2017) models. In both the approaches, the encoder is purposed to map a source sentence into a sequence of state vectors, whereas the decoder uses the previous decoder states, its last output, and the attention model state to infer the next target word (see Figure 2). In a broad sense, the attention mechanism selects and combines the encoder states that are most relevant to infer the next word (Luong, Pham, and Manning 2015). In our multi-lingual setting, the decoding process is triggered by specifying the target language identifier (Italian, *<it>*, in the example of Figure 2). In the following two sub-sections, we briefly summarize the main features of the two considered architectures.

3.1 Recurrent NMT

Recurrent NMT models employ recurrent neural networks (RNNs) to build the internal representations of both the encoder and decoder. Recurrent layers are in general implemented with LSTM (Hochreiter and Schmidhuber 1997) or GRU (Cho et al. 2014a) units, which include gates able to control the propagation of information over time. While the encoder typically uses a bi-directional RNN, so that both left-to-right and right-to-left word dependencies are captured (see left-hand of Figure 3), the decoder by design can only learn left-to-right dependencies. In general, deep recurrent NMT is achieved by stacking multiple recurrent layers inside both the encoder and the decoder.

While RNNs are in theory the most expressive type of neural networks (Siegelmann and Sontag 1995), they are in practice hard and slow to train. In particular, the combination of two levels of deepness, horizontal along time and vertical across the layers, makes gradient-based optimization of deep RNNs particularly slow to converge and difficult to parallelize (Wu et al. 2016). Recent work succeeded in speeding up training convergence (Sennrich et al. 2017a) of recurrent NMT by reducing the network size via parameter tying and layer normalization. On the other hand, the *simple recurrent* NMT model proposed by (Di Gangi and Federico 2018), which weakens the network time dependencies, has shown to outperform LSTM-based NMT both in training speed and performance.

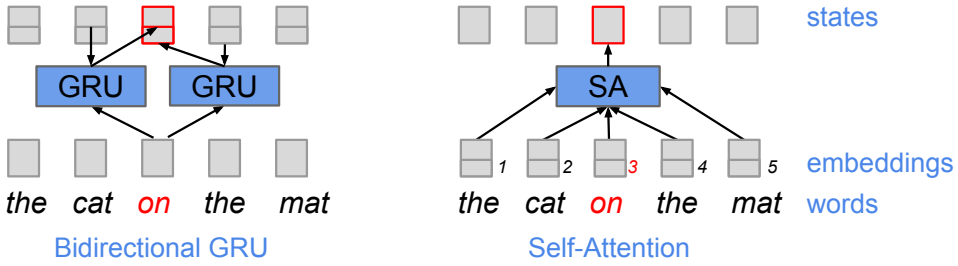

Figure 3
Single-layer encoders with recurrent (left) and transformer networks (right). A bi-directional recurrent encoder generates the state for word "on" with two GRU units. Notice that states must be generated sequentially. The transformer generates the state of word "on" with a self-attention model that looks at all the input embeddings, which are extended with position information. Notice that all the states can be generated independently.

3.2 Transformer NMT

The transformer architecture (Vaswani et al. 2017) works by relying on a self-attention mechanism, removing all the recurrent operations that are found in the RNN case (Vaswani et al. 2017). In other words, the attention mechanism is re-purposed to also compute the latent space representation of both the encoder and the decoder. The right-hand side of Figure 3 depicts a simple one-layer encoder based on self-attention. Notice that, in absence of recurrence, a *positional-encoding* is added to the input and output embeddings. Similarly, as the time-step in RNN, the positional information provides the transformer network with the order of input and output sequences. In our work we use absolute positional encoding but, very recently, the use of relative positional information has been shown to improve the network performance (Shaw, Uszkoreit, and Vaswani 2018).

Overall, the transformer is organized as a stack of encoder-decoder networks that works in an auto-regressive way, using the previously generated symbol as input for the next prediction. Both the decoder and encoder can be composed of uniform layers, each built of sub-layers, i.e., a multi-head self-attention sub-layer and a position-wise feed-forward network sub-layer. Specifically for the decoder, an extra multi-head attentional layer is added to attend to the output states of the encoder. Multi-head attention layers enable the use of multiple attention functions with a computational cost similar to utilizing a single attention.

4. Zero-Shot Self-Training in Multilingual NMT

In this setting, our goal is to improve translation in the zero-shot directions of a baseline multilingual model trained on data covering n languages but not all their possible combinations (see Figure 1). After training a baseline multilingual model with the target-forcing method (Johnson et al. 2017), our self-learning approach works in the following way:

- First, a dual zero-shot inference (i.e., source↔target directions) is performed utilizing monolingual data extracted from the training corpus;

- Second, the training resumes combining the inference output and the original multilingual data from the non zero-shot directions;
- Third, this cycle of *training-inference-training* is repeated until a convergence point is reached on the dual zero-shot directions.

Notice that, at each iteration, the original training data is augmented only with the last batch of generated translations. We observe that the generated outputs initially contain a mix of words from the shared vocabulary but, after few iterations, they tend to only contain words in the zero-shot target language thus becoming more and more suitable for learning. The training and inference strategy of the proposed approach is summarized in Algorithm 1, whereas the flow chart (see Figure 4) further illustrates the training and inference pipeline.

Table 1
Self-training algorithm for zero-shot directions $l_1 \leftrightarrow l_2$.

Algorithm 1: Train-Infer-Train (TIF)

1: TIF: (D, l_1, l_2)
2: $M \leftarrow$ Train($\emptyset$, D) //train multilingual base model on data D
3: $L_1 \leftarrow$ Extract(D, l_1) //extract l_1 monolingual data from D
4: $L_2 \leftarrow$ Extract(D, l_2) //extract l_2 monolingual data from D
5: **for** $i = 1, N$ **do**
6: $L_2^* \leftarrow$ Infer(M, L_1, l_2) //translate L_1 into l_2
7: $L_1^* \leftarrow$ Infer(M^*, L_2, l_1) //translate L_2 into l_1
8: $D^* \leftarrow D \cup (L_1^*, L_2) \cup (L_2^*, L_1)$ //augment original data
9: $M \leftarrow$ Train(M, D^*) //re-train model on augmented data
10: **end for**
11: return M

The proposed approach is performed in three steps, where the latter two are iterated for a few rounds. In the first step (line 2), a multilingual NMT system M is trained from scratch on the available data D ("*Train*" step). In the second step (lines 7-8), the last trained model M is run to translate ("*Infer*" step) between the zero-shot directions monolingual data L_1 and L_2 extracted from D (lines 3-4). Then, in the third step (line 10), training of M is re-started on the original data D plus the generated synthetic translations L_2^* and L_1^*, by keeping the extracted monolingual data L_1 and L_2 always on the target side ("*Train*" step). The updated model is then again used to generate synthetic translations, on which to re-train M, and so on.

In the multilingual NMT scenario, the automatic translations used as the source part of the extended training data will likely contain a mixed-language that includes words from a vocabulary shared with other languages. The expectation is that, round after round, the model will generate better outputs by learning at the same time to translate and "correct" its own translations by removing spurious elements from other languages. If this intuition holds, the iterative improvement will yield increasingly better results in translating between the source $\leftrightarrow$ target zero-shot directions.

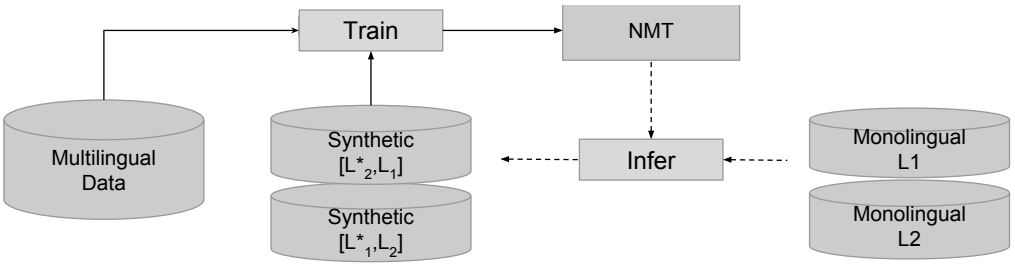

Figure 4
Illustration of the proposed multilingual *train-infer-train* strategy. Using a standard NMT architecture, a portion of two zero-shot directions monolingual dataset is extracted for inference to construct a dual source↔target mixed-input and continue the training. The solid lines show the training process, whereas the dashed lines indicate the inference stage.

5. Experiments

5.1 NMT Settings

We trained and evaluated multilingual NMT systems based on the RNN (Cho et al. 2014b) and transformer (Vaswani et al. 2017) models. Table 2 summarizes the hyper-parameters used for all our models. The RNN experiments are carried out using the NMT toolkit Nematus[2] (Sennrich et al. 2017b), whereas the transformer models are trained using the open source OpenNMT-tf[3] toolkit (Klein et al. 2017).

Training and inference hyper-parameters for both approaches and toolkits are fixed as follows. For the RNN experiments, the Adagrad (Duchi, Hazan, and Singer 2011) optimization algorithm is utilized with an initial learning rate of 0.01 and mini-batches of size 100. Considering the high data sparsity of our low-resource setting and to prevent model over-fitting (Srivastava et al. 2014), we applied a dropout on every layer, with probability 0.2 on the embeddings and the hidden layers, and 0.1 on the input and output layers. For the experiments using the transformer approach, a dropout of 0.3 is used globally. To train the baseline multilingual NMT, we use Adam (Kingma and Ba 2015) as the optimization algorithm with an initial learning rate scale constant of 2. For the transformer, the learning rate is increased linearly in the early stages (*warmup_training_steps*=$16,000$); after that, it is decreased with an inverse square root of training step (Vaswani et al. 2017).

In all the reported experiments, the baseline models are trained until convergence, while each training round after the inference stage is assumed to iterate for 3 epochs. In case of the transformer NMT, M4-NMT (four translation directions multilingual system), and M6-NMT (six translation directions multilingual system) BLEU scores are computed using averaged model from the last seven checkpoints in the same training run (Junczys-Dowmunt, Dwojak, and Sennrich 2016). For decoding, a beam search of size 10 is applied for recurrent models, whereas one of size 4 is used for transformer models.

2 https://github.com/EdinburghNLP/nematus
3 https://github.com/OpenNMT/OpenNMT-tf

Table 2
Hyper-parameters used to train RNN and transformer models, unless differently specified.

	enc/dec type	embedding size	hidden units	encoder depth	decoder depth	batch size
RNN	GRU	1024	1024	2	2	128 seg
Transformer	Transformer	512	512	6	6	4096 tok

5.2 Dataset and preprocessing

We run all our experiments on the multilingual translation shared task data released for the 2017 International Workshop on Spoken Language Translation (IWSLT)[4]. In particular, we used the subset of training data covering all possible language pair combinations between Italian, Romanian, and English (Cettolo, Girardi, and Federico 2012). For development and evaluation of the models, we used the corresponding sets from the IWSLT2010 (Paul, Federico, and Stüker 2010) and IWSLT2017 evaluation campaigns. Details about the used data sets are reported in Table 3. At the preprocessing stage, we applied word segmentation for each training condition (i.e. bilingual or multi-lingual) by learning a sub-word dictionary via Byte-Pair Encoding (Sennrich, Haddow, and Birch 2015b), by setting the number of merging rules to 39,500. We observe a rather high overlap between the language pairs (i.e the English dataset paired with Romanian is highly similar to the English paired with Italian). Because of this overlapping, the actual unique sentences in the dataset are approximately half of the total size. Consequently, on one side, this exacerbates the low-resource aspect in the multilingual models while, on the other side, we expect some positive effect on the zero-shot condition. The final size of the vocabulary, both in case of the bilingual and the multilingual models, stays under 40,000 sub-words. An evaluation script to compute the BLEU (Papineni et al. 2002) score is used to validate models on the dev set and later to choose the best performing models. Furthermore, significance tests computed for the BLEU scores are reported using Multeval (Clark et al. 2011).

Table 3
The total number of parallel sentences used for training, development, and test in our low-resource scenario.

Language Pair	*Train*	*Test10*	*Test17*
En-It	231619	929	1147
En-Ro	220538	929	1129
It-Ro	217551	914	1127

We trained models for two different scenarios. The first one is the multi-lingual scenario with all the available language pairs, while the second one is for the zero-shot and pivoting approaches which excludes $Romanian - Italian$ parallel sentences from the training data. For both scenarios, we have also trained bilingual RNN and Transformer

4 http://workshop2017.iwslt.org/

models for comparing bilingual against multilingual systems and for comparing pivoting with bilingual and multilingual models.

6. Models and Results

6.1 Bilingual Vs. Multilingual NMT

We compare the translation performance of six independently-trained bilingual models against one single multilingual model trained on the concatenation of all the six language pairs datasets, after prepending the language flag on the source side of each sentence. The performance of both types of systems is evaluated on test2017 and reported in Table 4. The experiments show that a multilingual system outperforms the bilingual systems with variable margins. The improvements, which are observed in all the language directions, are likely brought by the cross-lingual parameter transfer between the additional language pairs involved in the source and target side.

Table 4
Comparison between six bilingual models (NMT) against a single multilingual model (M6-NMT) on Test17.

Direction	*RNN*			*Transformer*		
	NMT	M6-NMT	Δ	NMT	M6-NMT	Δ
En → It	27.44	28.22	+0.78	29.24	30.88	+1.64
It → En	29.9	31.84	+1.94	33.12	36.05	+2.93
En → Ro	20.96	21.56	+0.60	23.05	24.65	+1.60
Ro → En	25.44	27.24	+1.80	28.40	30.25	+1.85
It → Ro	17.7	18.95	+1.25	20.10	20.13	+0.03
Ro → It	19.99	20.72	+0.73	21.36	21.81	+0.45

Table 4 shows that the transformer model is definitely superior to the RNN model for all directions and set-ups. With a larger margin of +3.22 (NMT) and +4.21 (M6-NMT), the transformer outperforms the RNN in the It-En direction. The closest performance between the two approaches is observed in the Ro-It direction, with the transformer showing a +1.37 (NMT) and +1.09 (M6-NMT) BLEU score increase compared to the RNN counterpart. Moreover, multilingual architectures in general outperform their equivalent models trained on single language pairs. The highest improvement of the M6-NMT over the NMT systems is observed when the target language is English. For instance, in the It-En direction, the multilingual approach gained +1.94 (RNN) and +2.93 (Transformer) over the single language pair models. Similarly, a +1.80 (RNN) and +1.85 (Transformer) gains are observed in the Ro-En direction. However, the smallest gain of the multilingual models occurred when translating into either Italian or Romanian. Independently from the target language in the experimental setting, the slight difference in the dataset size (that tends to benefit the English target, see Table 3) showed to impact the performance on non-English target directions.

6.2 Pivoting using a Multilingual Model

The pivoting experiment is setup by dropping the Italian↔Romanian parallel segments from the training data, and by training *i)* a new multilingual-model covering four

directions and *ii)* a single model for each language direction (It → En, En → It, Ro → En, En → Ro). Our main aim is to analyze how a multilingual model can improve a zero-shot translation task using a pivoting mechanism with English as a bridge language in the experiment. Moreover, the use of a multilingual model for pivoting is motivated by the results we acquired using the M6-NMT (see Table 4).

Table 5
Comparison of pivoting with bilingual models (NMT) and with multilingual models (M4-NMT) on Test17.

Direction	*RNN*			*Transformer*		
	NMT	M4-NMT	$\Delta^{\texttt{M4}-\texttt{NMT}}_{\texttt{NMT}}$	NMT	M4-NMT	$\Delta^{\texttt{M4}-\texttt{NMT}}_{\texttt{NMT}}$
It → En → Ro	16.3	17.58	+1.28	16.59	16.77	+0.18
Ro → En → It	18.69	18.66	-0.03	17.87	19.39	+1.52

The results in Table 5 show the potential, although partial, of using multilingual models for pivoting unseen translation directions. The comparable results achieved in both directions speak in favor of training and deploying one system instead of two distinct NMT systems. Remarkably, the marked difference between RNN and transformer is vanished in this condition. Pivoting using the M4-NMT system showed to perform better in three out of four evaluations, from the RNN and transformer runs. Note that the performance of the final translation (i.e pivot-target) is subject to the noise that has been propagated from the source-pivot translation step. Meaning pivoting is a favorable strategy when we have strong models in both directions of the pivot language.

6.3 Zero-shot Translations

For the direct zero-shot experiments and the application of the *train-infer-train* strategy, we only carried out experiments with the transformer approach. Preliminary results showed its superiority over the RNN together with the possibility to carry out experiments faster and with multiple GPUs.
In this experiment, we show how our approach helps to significantly boost the baseline multilingual NMT model. We run the train-infer-train for five consecutive stages, where each round consists in 2-3 epochs of additional training on the augmented training data. Table 6 shows the improvements on the dual Italian↔Romanian zero-shot directions.

Table 6
Comparison between a baseline multilingual model (M4-NMT) against the results from our proposed *train-infer-train* approach in a subsequent five rounds for the Italian↔Romanian zero-shot directions.

Direction	*M4-NMT*	*R1*	*R2*	*R3*	*R4*	*R5*
It→Ro	4.72	15.22	18.46	19.31	19.59	**20.11**
Ro→It	5.09	16.31	20.31	21.44	21.63	**22.41**

In both zero-shot directions the gain in a larger margin comes using the M-NMT model at *R1*. This is the first model trained after the inclusion of the dataset generated by the *dual-inference* stage. The It→Ro direction improves by $+10.50$ BLEU points from a 4.72 to 15.22, whereas Ro→It improves from a baseline score of 5.09 to 16.31 BLEU ($+11.22$).

The contribution of the self correcting process can be seen in the subsequent rounds, i.e., the improvements after each inference stage suggest that the generated data are getting cleaner and cleaner. With respect to the Transformer model pivoting results shown in Table 5, our approach outperformed both single pair and multilingual pivoting methods at the second round (*R2*) (see the third column of Table 6). Compared with the better performing multilingual pivoting, our approach at the fifth round (*R5*) has a $+3.34$ and $+3.02$ BLEU gain for the It→Ro and Ro→It directions respectively.

Table 7
Results summary comparing the performance of systems trained using parallel data (i.e., two single language pair *NMT* and a six direction multilingual *M6-NMT* systems) against the four directions multilingual baseline (*M4-NMT*) and our approach at the fifth round *R5*. Best scores are bold highlighted, whereas statistically significant ($p<0.05$) results in comparison with the baseline (*NMT*) are indicated with $\star$

Direction	*NMT*	*M6-NMT*	$\Delta^{\texttt{M6}-\texttt{NMT}}_{\texttt{NMT}}$	*M4-NMT*	$\Delta^{\texttt{M4}-\texttt{NMT}}_{\texttt{NMT}}$	*R5*	$\Delta^{\texttt{R5}}_{\texttt{NMT}}$
It→Ro	20.10	**20.13**	+0.03	4.72	-15.38	20.11	+0.01
Ro→It	21.36	21.81	+0.04	5.09	-16.27	**22.41***	+1.05

In addition to outperforming the pivoting mechanism, an interesting trend arises when we compare our approach with the results of the single language pair and multilingual models reported in Table 4. The summary in Table 7 shows the effectiveness of a dual-inference mechanism in allowing the model to learn from its outputs. Compared to the models trained using parallel data (i.e., *NMT* and *M6-NMT*), our approach (*R5*) is either comparable ($+0.01$ BLEU in It→Ro) or better performing ($+1.05$ BLEU in Ro→It). The trend from the *train-infer-train* stages indicates that, with additional rounds, our approach can further improve the dual translations. Overall, our iterative self-learning approach showed to deliver better results than the bilingual counterparts within five rounds, where each rounds iterates for a maximum of three epochs. Indeed, the improvement from our approach is a concrete example to train models in a self-learning way, potentially benefiting language directions with a parallel data, if casted in a similar setting.

7. Conclusions

In this paper, we used a multilingual NMT model in a low-resource language pairs scenario. Integrating and extending the work presented in (Lakew, Di Gangi, and Federico 2017) and (Lakew et al. 2017), we showed that a single multilingual system outperforms bilingual baselines while avoiding the need to train several single language pair models. In particular, we confirmed the superiority of transformer over recurrent NMT architectures in a multilingual setting. For enabling and improving a zero-shot translation, we showed *i)* how a multilingual pivoting can be used for achieving comparable results to those of multiple bilingual models, and *ii)* that our proposed self-learning procedure boosts performance of multilingual zero-shot directions by even outperforming both pivoting and bilingual models. In future work, we plan to explore our approach across language varieties using a multilingual model.

Acknowledgements

This work has been partially supported by the EC-funded projects ModernMT (H2020 grant agreement no. 645487) and QT21 (H2020 grant agreement no. 645452). This work was also supported by The Alan Turing Institute under the EPSRC grant EP/N510129/1 and by a donation of Azure credits by Microsoft. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

References

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations (ICLR)*, Alberta, Canada, April.

Béchara, Hanna, Yanjun Ma, and Josef van Genabith. 2011. Statistical post-editing for a statistical mt system. In *Machine Translation Summit XIII (MT Summit XIII)*, volume 13, pages 308–315, Beijing, China, September 19-23.

Bentivogli, Luisa, Arianna Bisazza, Mauro Cettolo, and Marcello Federico. 2018. Neural versus phrase-based mt quality: An in-depth analysis on english-german and english-french. *Computer Speech & Language*, 49:52–70.

Bertoldi, Nicola and Marcello Federico. 2009. Domain adaptation for statistical machine translation with monolingual resources. In *Proceedings of the fourth workshop on statistical machine translation*, pages 182–189, Athens, Greece, March. Association for Computational Linguistics.

Bojar, Ondrej, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, et al. 2016. Findings of the 2016 conference on machine translation (WMT16). In *Proceedings of the First Conference on Machine Translation (WMT)*, volume 2, pages 131–198, Berlin, Germany, August.

Cettolo, Mauro, Nicola Bertoldi, and Marcello Federico. 2011. Bootstrapping arabic-italian smt through comparable texts and pivot translation. In *15th Annual Conference of the European Association for Machine Translation (EAMT)*, Leuven, Belgium, May.

Cettolo, Mauro, Christian Girardi, and Marcello Federico. 2012. Wit^3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, pages 261–268, Trento, Italy, May.

Cettolo, Mauro, Jan Niehues, Sebastian Stüker, Luisa Bentivogli, Roldano Cattoni, and Marcello Federico. 2016. The IWSLT 2016 evaluation campaign. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, Seattle, WA, December 8-9.

Cho, Kyunghyun, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014a. On the properties of neural machine translation: Encoder-decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, Doha, Qatar, October, 25th.

Cho, Kyunghyun, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014b. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724 – 1734, Doha, Qatar, October.

Clark, Jonathan H., Chris Dyer, Alon Lavie, and Noah A. Smith. 2011. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 176–181, Portland, Oregon, USA, June. Association for Computational Linguistics.

Di Gangi, Mattia A. and Marcello Federico. 2018. Deep neural machine translation with weakly-recurrent units. In *Proceedings of the 21st Annual Conference of the European Association for Machine Translation (EAMT)*, Alicante, Spain, May. European Association for Machine Translation.

Dong, Daxiang, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1723–1732, Beijing, China, July.

Duchi, John, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159.

Firat, Orhan, Kyunghyun Cho, and Yoshua Bengio. 2016. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *Proceedings of NAACL-HLT*, San Diego, California, June.

Firat, Orhan, Baskaran Sankaran, Yaser Al-Onaizan, Fatos T. Yarman Vural, and Kyunghyun Cho. 2016. Zero-resource translation with multi-lingual neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 268–277, Austin, Texas, November.

Galley, Michel, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. 2006. Scalable inference and training of context-rich syntactic translation models. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 961–968, Sydney, Australia, July. Association for Computational Linguistics.

Ha, Thanh-Le, Jan Niehues, and Alexander Waibel. 2016. Toward multilingual neural machine translation with universal encoder and decoder. In *Proceedings of the 13th International Workshop on Spoken Language Translation (IWSLT)*, Seattle, USA, December.

He, Di, Yingce Xia, Tao Qin, Liwei Wang, Nenghai Yu, Tieyan Liu, and Wei-Ying Ma. 2016. Dual learning for machine translation. In *Advances in Neural Information Processing Systems*, pages 820–828, Barcelona, Spain, December.

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

Johnson, Melvin, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339 – 351.

Junczys-Dowmunt, Marcin, Tomasz Dwojak, and Rico Sennrich. 2016. The amu-uedin submission to the wmt16 news translation task: Attention-based nmt models as feature functions in phrase-based smt. In *Proceedings of the First Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 319 – 325, Berlin, Germany, August.

Kingma, Diederik and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the third International Conference for Learning Representations*, San Diego, California, May, 7-9.

Klein, Guillaume, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics-System Demonstrations*, pages 67 – 72, Vancouver, Canada, July-August.

Koehn, Philipp and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28 – 39, Vancouver, Canada, August,4.

Lakew, Surafel M., Mattia Antonino Di Gangi, and Marcello Federico. 2017. Multilingual neural machine translation for low resource languages. In *Proceedings of the 4th Italian Conference on Computational Linguistics (CLiC-it 2017)*, Rome, Italy, December.

Lakew, Surafel M., Quintino F. Lotito, Matteo Negri, Marco Turchi, and Marcello Federico. 2017. Improving zero-shot translation of low-resource languages. In *Proceedings of the 14th International Workshop on Spoken Language Translation*, Tokyo, Japan, December.

Luong, Minh-Thang, Quoc V. Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2016. Multi-task sequence to sequence learning. In *Proceedings of the 4th International Conference on Learning Representations (ICLR)*, San Juan, Puerto Rico, May 2-4.

Luong, Minh-Thang, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412 – 1421, Lisbon, Portugal, September 17-21.

Odlin, Terence. 1989. *Language Transfer: Cross-Linguistic Influence in Language Learning*. Cambridge University Press. Cambridge Books Online, June.

Oflazer, Kemal and Ilknur Durgar El-Kahlout. 2007. Exploring different representational units in english-to-turkish statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 25–32, Prague, Czech Republic, June. Association for Computational Linguistics.

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*, pages 311–318, NY, USA, July. Association for Computational Linguistics.

Paul, Michael, Marcello Federico, and Sebastian Stüker. 2010. Overview of the IWSLT 2010 evaluation campaign. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, Paris, France, December.

Sennrich, Rico, Alexandra Birch, Anna Currey, Ulrich Germann, Barry Haddow, Kenneth Heafield, Antonio Valerio Miceli Barone, and Philip Williams. 2017a. The university of edinburgh's neural mt systems for wmt17. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 389–399, Copenhagen, Denmark, September. Association for Computational Linguistics.

Sennrich, Rico, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Läubli, Antonio Valerio Miceli Barone, Jozef Mokry, et al. 2017b. Nematus: a toolkit for neural machine translation. In *Proceedings of the EACL 2017 Software Demonstrations*, pages 65 – 68, Valencia, Spain, April.

Sennrich, Rico, Barry Haddow, and Alexandra Birch. 2015a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 86 – 96, Berlin, Germany, August.

Sennrich, Rico, Barry Haddow, and Alexandra Birch. 2015b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1715 – 1725, Berlin, Germany, August.

Shaw, Peter, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proceedings of NAACL-HLT*, pages 464 – 468, New Orleans, Louisiana, June.

Siegelmann, Hava T. and Eduardo D. Sontag. 1995. On the computational power of neural nets. *Journal of Computer and System Sciences*, 50(1):132 – 150.

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

Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, Montréal, Canada, December.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010, Long Beach, USA, December.

Wu, Yonghui, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Zoph, Barret and Kevin Knight. 2016. Multi-source neural translation. In *Proceedings of NAACL-HLT 2016*, pages 30 – 34, San Diego, California, June.

Zoph, Barret, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568 – 1575, Austin, Texas, November.

Finding the Neural Net: Deep-learning Idiom Type Identification from Distributional Vectors

Yuri Bizzoni*
University of Gothenburg

Marco S. G. Senaldi**
Scuola Normale Superiore di Pisa

Alessandro Lenci†
Università di Pisa

The present work aims at automatically classifying Italian idiomatic and non-idiomatic phrases with a neural network model under constrains of data scarcity. Results are discussed in comparison with an existing unsupervised model devised for idiom type detection and a similar supervised classifier previously trained to detect metaphorical bigrams. The experiments suggest that the distributional context of a given phrase is sufficient to carry out idiom type identification to a satisfactory degree, with an increase in performance when input phrases are filtered according to human-elicited idiomaticity ratings collected for the same expressions. Crucially, employing concatenations of single word vectors rather than whole-phrase vectors as training input results in the worst performance for our models, differently from what was previously registered in metaphor detection tasks.

1. Introduction

Generally speaking, figurativeness has to do with pointing at a contextual interpretation for a given expression that goes beyond its mere literal meaning (Frege 1892; Gibbs et al. 1997; Cacciari and Papagno 2012). Let's imagine a commentator that, referring to an athlete, says *She's always delivered clean performances but this one really took the cake!* In this sentence, *clean performances* is an example of *metaphorical expression* that, according to the model proposed by Lakoff and Johnson (2008), reflects a rather transparent mapping between an abstract concept in a *target domain* (e.g., the flawlessness of a performance) and a concrete example taken from a *source domain* (e.g., the cleanliness of a surface). On the other hand, *take the cake* is an *idiom*, i.e. a lexicosyntactically rigid multiword unit (Sag et al. 2002) that is entirely non-compositional, since its meaning of 'being outstanding' is not accessible by simply composing the meanings of *take* and *cake* and must therefore be learnt by heart by speakers (Frege 1892; Cacciari 2014).

Important differences have been stressed between metaphors and idioms in theoretical (Gibbs 1993; Torre 2014), neurocognitive (Bohrn, Altmann, and Jacobs 2012) and corpus linguistic (Liu 2003) studies. First of all, metaphors represent a productive

* Department of Philosophy, Linguistics, Theory of Science - Dicksonsgatan 4, 41256, Göteborg, Sweden. E-mail: yuri.bizzoni@gu.se
** Scuola Normale Superiore - Piazza dei Cavalieri 7, I-56126 Pisa, Italy. E-mail: marco.senaldi@sns.it
† CoLing Lab, Department of Philology, Literature and Linguistics - Via S. Maria 36, I-56126 Pisa, Italy. E-mail: alessandro.lenci@unipi.it

phenomenon: studies on metaphor production strategies indeed show a large ability of language users to generalize and create new metaphors on the fly from existing ones, allowing researchers to hypothesize recurrent semantic mechanisms underlying a large number of productive metaphors (McGlone 1996; Lakoff and Johnson 2008). For example, starting from the *clean performance* metaphor above, we could also say the delivered performance was *neat*, *spick-and-span* and *crystal-clear* by sticking to the same conceptual domain of cleanliness. On the other hand, although most idioms originate as metaphors (Cruse 1986), they have undergone a crystallization process in diachrony, whereby they now appear as conventionalized and (mostly) fixed combinations that form a finite repository in a given language (Nunberg, Sag, and Wasow 1994). From a formal standpoint, though some idioms allow for restricted lexical variability (e.g., the concept of getting crazy can be conveyed both by *to go nuts* and *to go bananas*), this kind of variation is not as free and systematic as with metaphors and literal language (e.g., transforming the *take the cake* idiom above into *take the candy* would hinder a possible idiomatic reading) (Fraser 1970; Geeraert, Baayen, and Newman 2017). From the semantic point of view, it is interesting to observe how speakers can correctly use the most semantically opaque idioms in discourse without necessarily being aware of their actual metaphorical origin or anyway having contrasting intuitions about it. For example, Gibbs (1994) reports that many English speakers explain the idiom *kick the bucket* 'to die' as someone kicking a bucket to hang themselves, while it actually originates from a corruption of the French word *buquet* indicating the wooden framework that slaughtered hogs kicked in their death struggles. Secondly, metaphorical expressions can receive varying interpretations according to the context at hand: saying that *John is a shark* could mean that he's ruthless on his job, that he's aggressive or that he attacks people suddenly (Cacciari 2014). Contrariwise, idiomatic expressions always keep the same meaning: saying that *John kicked the bucket* can only be used to state that he passed away. Finally, idioms and metaphors differ in the mechanisms they recruit in language processing: while metaphors seem to bring into play *categorization* (Glucksberg, McGlone, and Manfredi 1997) or *analogical* (Gentner 1983) processes between the vehicle and the topic (e.g., *shark* and *John* respectively in the sentence above), idioms by and large call for lexical access mechanisms (Cacciari 2014). Nevertheless, it is crucial to underline that idiomaticity itself is a multidimensional and gradient phenomenon (Nunberg, Sag, and Wasow 1994; Wulff 2008) with different idioms showing varying degrees of semantic transparency, formal versatility, proverbiality and affective valence. All this variance within the class of idioms themselves has been demonstrated to affect the processing of such expressions in different ways (Cacciari 2014; Titone and Libben 2014).

The aim of this work is to focus on the fuzzy boundary between idiomatic and metaphorical expressions from a computational viewpoint, by applying a supervised method previously designed to discriminate metaphorical vs. literal usages of input constructions to the task of distinguishing idiomatic from compositional expressions. Our starting point is the work of Bizzoni, Chatzikyriakidis, and Ghanimifard (2017), who managed to classify adjective-noun pairs where the same adjectives were used both in a metaphorical and a literal sense (e.g., *clean performance* vs. *clean floor*) by means of a neural classifier trained on a composition of the words' embeddings (Mikolov et al. 2013). As the authors found out, the neural network succeeded in the task because it was able to detect the abstract/concrete semantic shift undergone by the nouns when used with the same adjective in figurative and literal compositions respectively. In our attempt, we will use a relatively similar approach to classify idiomatic expressions by training a three-layered neural network on a set of Italian idioms (e.g. *gettare la spugna* 'to

throw in the towel', lit. 'to throw the sponge') and non-idioms (e.g. *vedere una partita* 'to watch a match'). The performance of the network will be compared when trained with constructions belonging to different syntactic patterns, namely Adjective-Noun and Verb-Noun expressions (AN and VN henceforth). Noteworthily, the abstract/concrete polarity the network was able to learn in Bizzoni, Chatzikyriakidis, and Ghanimifard (2017) will not be available this time: while the nouns in the dataset of Bizzoni, Chatzikyriakidis, and Ghanimifard (2017) were used in their literal sense, idioms are entirely non-compositional, so none of their constituents is employed literally inside the expressions, independently of their concreteness (e.g., *spugna* 'sponge' in *gettare la spugna* vs *numeri* 'numbers' in *dare i numeri* 'to lose it', lit. 'to give the numbers'). What we want to find out is whether the sole information captured by the distributional vector of a given expression is sufficient for the network to learn its potential idiomaticity. The idiom classification scores of our models will be compared with those obtained by Senaldi, Lebani, and Lenci (2016) and Senaldi, Lebani, and Lenci (2017), who propose a distributional semantic algorithm for idiom type detection. Our study employs their small datasets. Therefore, the training sets we will operate on will be very scarce. Traditional ways to deal with data scarcity in computational linguistics resort to a wide number of different features to annotate the training set (see for example Tanguy et al. (2012)) or rely on artificial bootstrapping of the training set (He and Liu 2017). In our case, we test the performance of our classifier on scarce data without bootstrapping the dataset and relying only on the information provided by the distributional semantic space, showing that the distribution of an expression in large corpora can provide enough information to learn idiomaticity from few examples with a satisfactory degree of accuracy.

This paper is structured as follows: after reviewing in Section 2 the existing literature on idiom and metaphor processing, in Section 3 we will briefly outline the experimental design and in Section 4 we will provide details about the dataset we used and the human ratings we collected to validate our algorithms; in Section 5 we will go through the structure and functioning of our classifier and in Section 7 we will evaluate the performance of our models. Section 8 presents a qualitative error analysis, then followed by a discussion of the results (Section 9).

2. Related Work

Previous computational research has exploited different methods to perform *idiom type detection* (i.e., automatically telling apart potential idioms like *to get the sack* from only literal combinations like *to kill a man*). For example, Lin (1999) and Fazly, Cook, and Stevenson (2009) label a given word combination as idiomatic if the Pointwise Mutual Information (PMI) (Church and Hanks 1991) between its constituents is higher than the PMIs between the components of a set of lexical variants of this combination obtained by replacing the component words of the original expressions with semantically related words. Other studies have resorted to Distributional Semantics (Lenci 2008, 2018; Turney and Pantel 2010) by measuring the cosine between the vector of a given phrase and the single vectors of its components (Fazly and Stevenson 2008) or between the phrase vector and the sum or product vector of its components (Mitchell and Lapata 2010; Krčmář, Ježek, and Pecina 2013). Senaldi, Lebani, and Lenci (2016) and Senaldi, Lebani, and Lenci (2017) combine insights from both these approaches. They start from two lists of 90 VN and 26 AN constructions, the former composed of 45 idioms (e.g., *gettare la spugna*) and 45 non-idioms (e.g., *vedere una partita*), the latter comprising 13 idioms (e.g., *filo rosso* 'common thread', lit. 'red thread') and 13 non-idioms (e.g., *lungo periodo*

'long period'). For each of these constructions, a series of lexical variants are generated distributionally or via MultiWordNet (Pianta, Bentivogli, and Girardi 2002) by replacing the subparts of the constructions with semantically related words (e.g. from *filo rosso*, variants like *filo nero* 'black thread', *cavo rosso* 'red cable' and *cavo nero* 'black cable' are generated). What comes to the fore is that the vectors of the idiomatic expressions are less similar to the vectors of their lexical variants with respect to the similarity between the vector of a literal constructions and the vectors of its lexical alternatives. To provide an example, the cosine similarity between the vector of an idiom like *filo rosso* and the vectors of its lexical variants like *filo nero* and *cavo rosso* was found to be smaller than the cosine similarity between the vector of a literal phrase like *lungo periodo* and the vectors of its variants like *interminabile tempo* 'endless time' and *breve periodo* 'short period'.

Moving to the methodology exploited in the current study, to the best of our knowledge, neural networks have been previously adopted to perform MWE detection in general (Legrand and Collobert 2016; Klyueva, Doucet, and Straka 2017), but not idiom identification specifically. As mentioned in the Introduction, in Bizzoni, Chatzikyriakidis, and Ghanimifard (2017), pre-trained noun and adjective vector embeddings are fed to a single-layered neural network to disambiguate metaphorical and literal AN combinations. Several combination algorithms are experimented with to concatenate adjective and noun embeddings. All in all, the method is shown to outperform the state of the art, presumably leveraging the abstractness degree of the noun as a clue to figurativeness and basically treating the noun as the "context" to discriminate the metaphoricity of the adjective (cf. *clean performance* vs *clean floor*, where *performance* is more abstract than *floor* and therefore the mentioned cleanliness is to be intended metaphorically).

Besides Bizzoni, Chatzikyriakidis, and Ghanimifard (2017), using neural networks for metaphor detection with pretrained word embeddings initialization has been tried in a small number of recent works, proving that this is a valuable strategy to predict metaphoricity in datasets. Rei et al. (2017) present an ad-hoc neural design able to compose and detect metaphoric bigrams in two different datasets. Do Dinh and Gurevych (2016) apply a series of perceptrons to the VU Amsterdam Metaphor Corpus (Steen et al. 2014) combined with word embeddings and part-of-speech tagging. Finally, a similar approach - a combination of fully connected networks and pre-trained word embeddings - has also been used as a pre-processing step to metaphor detection, in order to learn word and sense abstractness scores to be used as features in a metaphor identification pipeline (Köper and Schulte im Walde 2017).

3. Method

In this work we carried out a supervised idiom type identification task by resorting to a three-layered neural network classifier. After selecting our dataset of VN and AN target expressions (Section 4.1), for which gold standard idiomaticity ratings had already been collected (Section 4.2), we built count vector representations for them (Section 4.3) from the itWaC corpus (Baroni et al. 2009) and fed them to our classifier (Section 5) with different training splits (Section 6). The network returned a binary output, whereby idioms were taken as our positive examples and non-idioms as our negative ones. Differently from Bizzoni, Chatzikyriakidis, and Ghanimifard (2017), for each idiom or non-idiom we initially built a count-based vector (Turney and Pantel 2010) of the expression as a whole, taken as a single token. We then compared this approach with a model trained on the concatenation of the individual words of an expression, but the latter turned out to be less effective for idioms than for metaphors. Each model was finally evaluated

in terms of classification accuracy, ranking performance and correlation between its continuous scores and the human-elicited idiomaticity judgments (Section 7).

Since we mostly worked with vectors that took our target expressions as unanalyzed wholes, as if they were single tokens, we were not concerned with the fact that some verbs were shared by more than one idiom (e.g., *lasciare il campo* 'to leave the field' and *lasciare il segno* 'to leave one's mark') or non-idiom (e.g., *andare a casa* 'to go home' and *andare all'estero* 'to go abroad') at once, given that our network could not access this information.

4. Dataset

4.1 Target expressions selection

The two datasets we employed in the current study come from Senaldi, Lebani, and Lenci (2016) and Senaldi, Lebani, and Lenci (2017). The first one is composed of 45 idiomatic Italian V-NP and V-PP constructions (e.g., *tagliare la corda* 'to flee' lit. 'to cut the rope') that were selected from an Italian idiom dictionary (Quartu 1993) and extracted from the itWaC corpus (Baroni et al. (2009), 1,909M tokens ca.) and whose frequency spanned from 364 (*ingannare il tempo* 'to while away the time') to 8294 (*andare in giro* 'to get about'), plus other 45 Italian non-idiomatic V-NP and V-PP constructions of comparable frequencies (e.g., *leggere un libro* 'to read a book'). The latter dataset comprises 13 idiomatic and 13 non-idiomatic AN constructions (e.g., *punto debole* 'weak point' and *nuova legge* 'new law') that were still extracted from itWaC and whose frequency varied from 21 (*alte sfere* 'high places', lit. 'high spheres') to 194 (*punto debole*).

4.2 Gold standard idiomaticity judgments

Senaldi, Lebani, and Lenci (2016) and Senaldi, Lebani, and Lenci (2017) collected gold standard idiomaticity judgments for the 26 AN and 90 VN target constructions in their datasets. Nine linguistics students were presented with a list of the 26 AN constructions and were asked to evaluate how idiomatic each expression was from 1 to 7, with 1 standing for 'totally compositional' and 7 standing for 'totally idiomatic'. Inter-coder agreement, measured with Krippendorff's α (Krippendorff 2012), was equal to 0.76. The same procedure was repeated for the 90 VN constructions, but in this case the inital list was split into 3 sublists of 30 expressions, each one to be rated by 3 subjects. Krippendorff's α was 0.83 for the first sublist and 0.75 for the other two. These inter-coder agreement scores were taken as a confirmation of reliability for the collected ratings (Artstein and Poesio 2008). As will become clear in Section 6, these judgments served the twofold purpose of evaluating the classification performance of our neural network and filtering the expressions to use as training input for our models.

4.3 Building target vectors

Count-based Distributional Semantic Models (DSMs) (Turney and Pantel 2010) allow for representing words and expressions as high-dimensionality vectors, where the vector dimensions register the co-occurrence of the target words or expressions with some contextual features, e.g. the content words that linearly precede and follow the target element within a fixed contextual window. We trained two DSMs on itWaC, where our target AN and VN idioms and non-idioms were represented as target vectors and co-occurrence statistics counted how many times each target construction occurred in the

same sentence with each of the 30,000 top content words in the corpus. Differently from Bizzoni, Chatzikyriakidis, and Ghanimifard (2017), we did not opt for prediction-based vector representations (Mikolov et al. 2013). Although some studies have brought out that context-predicting models fare better than count-based ones on a variety of semantic tasks (Baroni, Dinu, and Kruszewski 2014), including compositionality modeling (Rimell et al. 2016), others (Blacoe and Lapata 2012; Cordeiro et al. 2016) have shown them to perform comparably. In phrase similarity and paraphrase tasks, Blacoe and Lapata (2012) find count vectors to score better then or comparably to predict vectors built following Collobert and Weston (2008)'s neural language model. Cordeiro et al. (2016) show PPMI-weighted count-based models to perform comparably to *word2vec* (Mikolov, Yih, and Zweig 2013) in predicting nominal compound compositionality. Moreover, Levy, Goldberg, and Dagan (2015) highlight that much of the superiority in performance exhibited by word embeddings is actually due to hyperparameter optimizations, which, if applied to traditional models as well, can bring to equivalent outcomes. Therefore, we felt confident in resorting to count-based vectors as an equally reliable representation for the task at hand.

5. The neural network classifier

We built a neural network composed of three "dense" or fully connected hidden layers.[1] The input layer has the same dimensionality of the original vectors and the output layer has dimensionality 1. The other two hidden layers have dimensionality 12 and 8 respectively. Our network takes in input a single vector at a time, which can be a word embedding, a count-based distributional vector or a composition of several word vectors. For the core part of our experiment we used as input single distributional vectors of two-word expressions. As we discussed in the previous section, these vectors have 30,000 dimensions each and represent the distributional behavior of a full expression rather than that of the individual words composing such expression. Given this distributional matrix, we defined idioms as positive examples and non-idioms as negative examples of our training set. Due to the magnitude of our input, the most important reduction of data dimensionality is carried out by the first hidden layer of our model. The last layer applies a sigmoid activation function on the output in order to produce a binary judgment. While binary scores are necessary to compute the model classification accuracy and will be evaluated in terms of F1, our model's continuous scores can be retrieved and will be used to perform an ordering task on the test set, that we will evaluate in terms of Interpolated Average Precision (IAP) [2] and Spearman's ρ with the human-elicited idiomaticity judgments. IAP and ρ, therefore, will be useful to investigate how good our model is in ranking idioms before non-idioms.

6. Choosing the training set

The scarcity of our training sets constitutes a challenge for neural models, typically designed to deal with massive amounts of data. The typical effect of such scarcity is a fluctuation in performance: training our model on two different sections of the same dataset is likely to result in quite different F-scores.

1 We used Keras, a library running on TensorFlow (Abadi et al. 2016).
2 Following Fazly, Cook, and Stevenson (2009), IAP was computed at recall levels of 20%, 50% and 80%.

Unless otherwise specified, the IAP, Spearman's ρ and F1 scores reported in Table 1 are averaged on 5 runs of each model on the same datasets: at each run, the training split is randomly selected. We found that some samples of the training set seemingly make it harder for the model to learn idiom detection. When such runs are included in the mean, the performance is drastically lowered.

In our attempt to understand whether we could find a rationale behind this phenomenon or it was instead completely unpredictable, in some versions of our models we have tried to filter our training sets according to the idiomaticity judgments we elicited from speakers (Section 4.2) to assess which composition of our training sets made our algorithm more effective. In the first approach, which we will label as High-to-Low (HtL henceforth), the network was trained on the idioms receiving the highest idiomaticity ratings (and symmetrically on the compositional expressions having the lowest idiomaticity ratings) and was therefore tested on the intermediate cases. In the second approach, which we called Low-to-High (LtH), the model was trained on more borderline exemplars, i.e. the idioms having the lowest idiomaticity ratings and the compositional expressions having the highest ones, and then tested on the most polarized cases of idioms and non-idioms.

For example, in the HtL setting, the AN bigrams we selected for the training set included idioms like *testa calda* 'hothead' and *faccia tosta* 'brazen person' (lit. 'tough face'), that reported an average idiomaticity rating of 6.8 and 6.6 out of 7 respectively, and non-idioms like *famoso scrittore* 'famous writer' and *nuovo governo* 'new government' that elicited an average idiomaticity rating of 1.2 and 1.1 out of 7. In the case of VN bigrams, we selected idioms like *andare a genio* 'to sit well' (lit. 'to go to genius') (mean idiomaticity rating of 7) and non-idioms like *vendere un libro* 'to sell a book' (mean idiomaticity rating of 1). The neural network was thus trained only on elements that our annotators had judged as clearly positive and clearly negative examples.

To provide examples on the LtH training sets, for the VN data, we selected idioms like *lasciare il campo* (mean rating = 3.6) and *cambiare colore* 'to change color (in face)' (mean rating = 3.6) against non-idiomatic expressions like *prendere un caffè* 'to grab a coffee' (3.3) and *lasciare un incarico* 'to leave a job' (2.3). For the AN data, we selected idioms like *prima serata* 'prime time' (lit. 'first evening') (mean rating = 4 out of 7) and compositional expressions like *proposta concreta* 'concrete proposal' (2.7). The neural network was in this case trained only on elements that our annotators had judged as borderline cases.

The results of these different filtering procedures can be found in Table 1.

7. Evaluation

Once the training sets were established, a variety of transformations were tried on our VN and AN distributional vectors before giving them as input to our network. Some models were trained on the raw 30,000 dimensional distributional vectors of VN and AN expressions; other models used the concatenation of the vectors of the individual components of the expressions; finally, other models employed PPMI (Positive Pointwise Mutual Information) (Church and Hanks 1991) and SVD (Singular Value Decomposition) transformed (Deerwester et al. 1990) vectors of 150 and 300 dimensions. Details of both classification and ordering tasks are shown in Table 1. Qualitative details about the results will be given in Section 8.

Table 1
Interpolated Average Precision (IAP), Spearman's ρ correlation with the human judgments and F-measure (F1) for Vector-Noun training (VN), Adjective-Noun training (AN), joint (VN+AN) training and training through vector concatenation. High-to-Low (HtL) models were trained on clear-cut cases, while Low-to-High (LtH) models were trained on borderline cases. As for the other models, the average performance over 5 runs with randomly selected training sets is reported. Training and test set are expressed as the sum of positive and negative examples.

Vectors	PPMI	SVD	Training	Test	IAP	ρ	F1
VN	Yes	No	15+15	30+30	.72	.48	.67
VN	Yes	No	20+20	25+25	**.73**	**.52**	**.77**
VN	Yes	150	15+15	30+30	.63	.35	.48
VN	Yes	150	20+20	25+25	.61	.33	.63
VN	Yes	300	15+15	30+30	.67	.33	.64
VN	Yes	300	20+20	25+25	.65	.3	.57
AN	No	No	8+8	6+4	.72	.19	.40
AN	Yes	No	8+8	6+4	.70	.06	**.60**
AN	Yes	150	8+8	6+4	.65	.11	.32
AN	Yes	300	8+8	6+4	**.88**	**.51**	.10
VN (HtL)	Yes	No	15+15	30+30	.71	.62	.77
VN (HtL)	Yes	No	20+20	25+25	**.79**	.65	.84
VN (LtH)	Yes	No	15+15	30+30	.71	.58	.80
VN (LtH)	Yes	No	20+20	25+25	.77	**.68**	**.85**
AN (HtL)	No	No	8+8	6+4	1	.8	.71
AN (HtL)	Yes	No	8+8	6+4	1	.71	.78
AN (LtH)	No	No	8+8	6+4	1	**.93**	**.89**
AN (LtH)	Yes	No	8+8	6+4	1	.84	.88
VN+AN	No	No	23+23	36+34 (joint)	.80	.64	.46
VN+AN (HtL)	No	No	23+23	36+34 (joint)	.63	.41	.65
VN+AN (LtH)	No	No	23+23	36+34 (joint)	**.68**	**.51**	**.66**
Conc. VN	No	No	20+20	24+24	.59	.34	.40
Conc. VN (HtL)	No	No	20+20	24 +24	**.61**	.07	.46
Conc. VN (LtH)	No	No	20+20	24+24	.57	**.31**	**.59**

7.1 Verb-Noun

We ran our model on the VN dataset, composed of 90 elements, namely 45 idioms and 45 non-idiomatic expressions. This is the largest of the two datasets. We trained our model on 30[3] and 40 elements for 20 epochs and tested it on the remaining 60 and 50 elements respectively. The models that best succeeded at classifying our phrases into idioms and non-idioms were trained with 40 PPMI-transformed vectors, reaching an average F1 score of .77 on the randomized iterations and an F1 score of .85, with a Spearman's ρ correlation of .68, when the training set was composed of borderline cases and the model was then tested on more clear-cut exemplars (LtH). As for the rest of the F1 scores,

3 When we report the number of training and test items in Table 1 as 15+15, for instance, we mean 15 idioms + 15 non-idioms. The same applies to all the other listed models.

what comes to light from our results is that increasing the number of training vectors generally leads to better results, except for models fed with SVD-transformed vectors of 300 dimensions, which seem to be insensitive to the size of our training data. Quite interestingly, SVD-reduced vectors appear to perform worse in general than raw ones and just PPMI-transformed ones. Due to space limitations, raw-frequency VN models are not reported in Table 1 since they were comparable to just PPMI-weighted ones.

This same pattern is encountered when evaluating the ability of our algorithm to rank idioms before non-idioms (IAP). The models with the highest score employs 40 PPMI training vectors and reach .73 on the randomized training, .79 on the HtL training and .77 on the LtH ones, while SVD training vectors generally lead to poorer ranking performances. Despite these IAP scores being encouraging, they are anyway lower than those obtained by Senaldi, Lebani, and Lenci (2016), who reach a maximum IAP of 0.91. This drop in performance could point to the fact that resorting to distributional information only to carry out idiom identification overlooks some aspects of the behavior of idiomatic constructions (e.g., formal rigidity) that is to be taken into account to arrive at a more satisfactory classification. Concerning the correlation between the continuous score of the neural net and the human idiomaticity ratings presented in Section 4.2, the best model also employed 40 PPMI vectors of borderline expressions (.68), followed by the model using 40 PPMI vectors of clear-cut cases (.65). These correlation values are quite comparable to the maximum of -0.67 obtained in Senaldi, Lebani, and Lenci (2016)[4] in High-to-Low and Low-to-High ordered models, while they are lower in randomized models, especially SVD-reduced ones.

All in all, both HtL and LtH experimental settings result in IAP, correlation and F1 scores that are higher than what we get from averaging over randomly selected training sets. More precisely, the strategy of training only on borderline examples (LtH) appears to be the most effective. This can intuitively make sense: once a network has learned to discriminate between borderline cases, detecting clear-cut elements should be relatively easy. The opposite strategy also seems to bring some benefits, possibly because training on clear negative and positive examples provides the network with a data set which is easier to generalize. In any case, it seems clear that selecting our training set with the help of human ratings allows us to significantly increase the performance of our models. We can see this as another proof that human continuous scores on idiomaticity - and not only binary judgments - are mirrored in the distributional pattern on these expressions. As for the influence of the training set size on IAP and ρ, all in all it seems that the best results are reached with 40 training vectors, both on the randomized training sets and on the ordered training sets.

The general trend we can abstract from these results is that our neural network does a good job in telling apart idioms and non-idioms by just relying on raw-frequency and PPMI-transformed distributional information. Performing dimensionality reduction apparently deprives the model of useful information, which makes the overall performance plummet to lower levels.

4 Please keep in mind that the correlation values in Senaldi, Lebani, and Lenci (2016) and Senaldi, Lebani, and Lenci (2017) are negative since the less similar a target vector to the vectors of its variants, the more idiomatic the target.

7.2 Adjective-Noun

Our model was also run on the AN dataset, composed of 26 elements (13 idioms and 13 non-idiomatic expressions). We empirically found that our network was able to perform some generalization on the data when the training set contained at least 14 elements, evenly balanced between positive and negative examples. We trained our model on 16 elements for 30 epochs and tested on the remaining 10 elements. As happened with VN vectors, performing SVD worsened the performance of the model. While F1 exact value can undergo fluctuations when a model is trained on very small sets, we always registered accuracies higher than 70% for the ordered training sets. In this case even more than in the Verb-Noun frame, the difference between randomizing the training set and selecting it using human idiomaticity ratings appears to be very evident, possibly due to the extremely small dimensions of this specific dataset, that make the qualitative selection of the training data of particular importance. Once again the highest Spearman's ρ correlation (.93) was reached when using a Low-to-High set trained on borderline cases, although it is important to keep in mind that such scores are computed on a very restricted test set. The same reasoning applies to IAP scores, which all reach the top value, though we must consider the very small test set. Senaldi, Lebani, and Lenci (2017) instead reached a maximum IAP of .85 and a maximum ρ of -.68 in AN idiom identification. When the training size was under the critical threshold, accuracy dropped significantly. With training sets of 10 or 12 elements, our model naturally went in overfitting, quickly reaching 100% accuracy on the training set and failing to correctly classify unseen expressions. In these cases a partial learning was still visible in the ordering task, where most idioms, even if labeled incorrectly, received higher scores than non-idioms.

7.3 Joint training

Our last experiment consisted in training our model on a mixed dataset of both VN and AN expressions, to check to what extent it would be able to recognize the same underlying semantic phenomenon across different syntactic constructions. In these models as well as in those described in Section 7.4, PPMI and SVD transformations were not tested anymore, since they were already shown to bring to generally comparable or even worse outcomes when tried on the VN and the AN datasets singularly. Concerning the structure of our training and test sets, two approaches were experimented with. We first tried to train our model on one pair type, e.g. the AN pairs, and then tested on the other, but we saw this required more epochs overall (more than 100) to stabilize and resulted in a poorer performance. When training our model on a mixed dataset containing the elements of both pair types, our model employed 20 epochs to reach an F-measure of 66% on the mixed training set when the set was ordered Low-to-High (i.e., it was composed of borderline cases only) and a comparable F-score of 65% when using clear-cut training input (HtL). Anyway, we also noticed that VN expressions were learned better than AN expressions. It's also worth considering that, although the F-scores of the LtH and HtL models were higher, the IAP and Spearman's ρ were lower than in the unordered input model. In other words, while ordering the input led to a better binary classification, the continuous scores returned a less precise ranking.

Our model was able to generalize over the two datasets, but this involved a loss in accuracy with respect to the only-VN and only-AN ordered training sets. It can be seen in Table 1 that a loss in accuracy is also evident for joint training on the randomized frame, although in this case the model seems hardly able to generalize at all.

7.4 Vector concatenation

In addition to using the vector of an expression as a whole, we tried to feed our model with the concatenation of the vectors of the single words in an expression, as in Bizzoni, Chatzikyriakidis, and Ghanimifard (2017). For example, instead of using the 30,000 dimensional vector of the expression *tagliare la corda*, we used the 60,000 dimensional vector resulting from the concatenation of *tagliare* and *corda*. This approach mimics the one adopted for metaphoric pairs and concludes our set of experiments, providing us with comparable results obtained from a compositionality-based approach to the same problem. We ran this experiment only on the VN dataset, being the largest and the one that yielded the best results in the previous settings. We used 40 elements in training and 48 in testing and trained our model for 30 epochs overall. Predictably enough, vector composition resulted in the worst performance, differently from what happened with metaphors (Bizzoni, Chatzikyriakidis, and Ghanimifard 2017).

Despite all correlations are low and not statistically significant, it is still worth pointing out however that not all the results are completely random: with an F1 of 59% for the LtH training set and an IAP of .61 for the HtL set, the model seems able to learn idiomaticity to a lower, but not null, degree; these findings would be in line with the claim that the meaning of the subparts of several idioms, while less important than in metaphors, is not completely obliterated (McGlone, Glucksberg, and Cacciari 1994). Another hint in this direction is the difference in performance between randomized and ordered training that we can observe for concatenation: if human idiomaticity ratings were completely independent from the composition of the individual subparts of our idioms, such effect should not be present at all. Anyway, similarly to what happened with the joint models, ordering the training input led to higher F-scores and comparable IAPs, but returned a worse correlation with human judgments with respect to the models with a randomized training input.

8. Error Analysis

As we mentioned in Section 1, idiomaticity is not a black-or-white phenomenon and idioms are rather spread on a continuum of semantic transparency and formal rigidity, which makes some exemplars harder to classify. In our models we can find some "prototypical" cases of idioms which were always labeled correctly, like *toccare il fondo* 'to hit rock bottom', *lasciare il campo* and *passare alla storia* 'to go down in history' and also some cases of unambiguously classified non-idioms, like *andare in vacanza* 'to go on holiday', *ascoltare una canzone* 'to listen to a song' and *prendere un caffè*. On the other hand, we have some ambiguous expressions like *abbassare la guardia* 'to let down one's guard' and *sentire una voce* 'to hear a voice', which, despite being compositional and potentially literal, can be very often used figuratively, i.e. if someone were referring to *guardia* as a metaphorical defense or to *voce* as a rumor. In such cases, it might be the case that the evidence available in the chosen corpus privileged just one of the two possible readings, leading to labeling issues. By the same token, the expression *bussare alla porta (di qualcuno)* 'to go ask for (someone's) help' (lit.'to knock at the door'), which we initially labeled as idiomatic, can have a literal reading as well and that is why it was often labeled as non-idiomatic. Finally, as happened in Senaldi, Lebani, and Lenci (2016), some false positives like *chiedere le dimissioni* 'to demand the resignation' and *entrare in crisi* 'to get into a crisis' are compositional expressions which nonetheless display collocational behavior, since they represent very common and fixed expressions in the Italian language. Interestingly, while Senaldi, Lebani, and Lenci (2016) could

justify their being false positives since it is likely that the variant-based model took their lexical fixedness as a clue of their idiomatic status, our neural net relies on distributional semantic information only. What this suggests is that not only a semantic phenomenon like compositionality, but even a shallower one like collocability, which does not always and straightforwardly go hand in hand with non-compositionality, can be spotted out just by looking at contextual distribution.

As mentioned in Section 4.1, our target idioms and non-idioms varied considerably in frequency. We therefore conducted some correlation analyses to check out a possible relationship between the scores returned by our network and the frequency of our items. All in all, we can conclude that in most of our models frequency and the continuous idiomaticity scores were negatively correlated, though such a correlation did not show up systematically and was not always significant. In other words, the more frequent an item, be it an idiom or a literal, the more the network tended to consider it as literal (i.e., it gave it a lower idiomaticity score). This tendency could be explained if we consider that some of our most frequent idioms were actually quite ambiguous (e.g., *aprire gli occhi* 'to open one's eyes' occurred 6306 times in the corpus and *bussare alla porta* 3303 times) and most of their corpus occurrences could be literal uses.

9. Discussion and Conclusions

The experiments we have presented show that the distribution of idiomatic and compositional expressions in large corpora can suffice for a supervised classifier to learn the difference between the two linguistic elements from small training sets and with a good level of accuracy. Specifically, we have observed that human continuous ratings of idiomaticity can be useful to select a better training set for our models, and that training our models on cases deemed by our annotators as borderline allows them to learn and perform better than if they were fed with randomized input. Also training our models only on clear-cut cases increases the performance. In general we can see from this phenomena that human continuous ratings of idiomaticity seem to be mirrored in the distributional structure of our data.

Unlike with metaphors (Bizzoni, Chatzikyriakidis, and Ghanimifard 2017), feeding the classifier with a composition of the individual words' vectors of such expressions performs quite scarcely and can be used to detect only some idioms. This takes us back to the core difference that while metaphors are more compositional and preserve a transparent source domain to target domain mapping, idioms are by and large non-compositional. Since our classifiers rely only on contextual features, their ability in classification must stem from a difference in distribution between idioms and non-idioms. A possible explanation is that while the literal expressions we selected, like *vedere un film* or *ascoltare un discorso*, tend to be used with animated subjects and thus to appear in more concrete contexts, most of our idioms (e.g. *cadere dal cielo* or *lasciare il segno*) allow for varying degrees of animacy or concreteness of the subject, and thus their context can easily get more diverse. At the same time, the drop in performance we observe in the joint models seems to indicate that the different parts of speech composing our elements entail a significant contextual difference between the two groups, which introduces a considerable amount of uncertainty in our model.

It is also possible that other contextual elements we did not consider have played a role in the learning process of our models, like the ambiguity between idiomatic and literal meaning that some potentially idiomatic strings possess (e.g. *to leave the field*) and that would lead their contextual distribution to be more variegated with respect to only-literal combinations. We intend to further investigate this aspect in future works.

References

Abadi, Martín, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*.

Artstein, Ron and Massimo Poesio. 2008. Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596.

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

Baroni, Marco, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 238–247, Baltimore, USA, June 22-27.

Bizzoni, Yuri, Stergios Chatzikyriakidis, and Mehdi Ghanimifard. 2017. "deep" learning: Detecting metaphoricity in adjective-noun pairs. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark, September 7-11.

Blacoe, William and Mirella Lapata. 2012. A comparison of vector-based representations for semantic composition. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 546–556, Jeju Island, Korea, July 12-14. Association for Computational Linguistics.

Bohrn, Isabel C., Ulrike Altmann, and Arthur M. Jacobs. 2012. Looking at the brains behind figurative language: a quantitative meta-analysis of neuroimaging studies on metaphor, idiom, and irony processing. *Neuropsychologia*, 50(11):2669–2683.

Cacciari, Cristina. 2014. Processing multiword idiomatic strings: Many words in one? *The Mental Lexicon*, 9(2):267–293.

Cacciari, Cristina and Costanza Papagno. 2012. Neuropsychological and neurophysiological correlates of idiom understanding: How many hemispheres are involved. *The handbook of the neuropsychology of language*, pages 368–385.

Church, Kenneth W. and Patrick Hanks. 1991. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29.

Collobert, Ronan and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167, Helsinki, Finland, July 5-9. ACM.

Cordeiro, Silvio, Carlos Ramisch, Marco Idiart, and Aline Villavicencio. 2016. Predicting the compositionality of nominal compounds: Giving word embeddings a hard time. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 1986–1997, Berlin, Germany, August 7-12.

Cruse, D. Alan. 1986. *Lexical semantics*. Cambridge University Press.

Deerwester, Scott, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391.

Do Dinh, Erik-Lân and Iryna Gurevych. 2016. Token-level metaphor detection using neural networks. In *Proceedings of the Fourth Workshop on Metaphor in NLP*, pages 28–33, San Diego, USA, June 17.

Fazly, Afsaneh, Paul Cook, and Suzanne Stevenson. 2009. Unsupervised type and token identification of idiomatic expressions. *Computational Linguistics*, 1(35):61–103.

Fazly, Afsaneh and Suzanne Stevenson. 2008. A distributional account of the semantics of multiword expressions. *Italian Journal of Linguistics*, 1(20):157–179.

Fraser, Bruce. 1970. Idioms within a transformational grammar. *Foundations of language*, pages 22–42.

Frege, Gottlob. 1892. Über sinn und bedeutung. *Zeitschrift für Philosophie und philosophische Kritik*, 100:25–50.

Geeraert, Kristina, R. Harald Baayen, and John Newman. 2017. Understanding idiomatic variation. In *Proceedings of the 13th Workshop on Multiword Expressions*, page 80, Valencia, Spain, April 4.

Gentner, Dedre. 1983. Structure-mapping: A theoretical framework for analogy. *Cognitive science*, 7(2):155–170.

Gibbs, Raymond W. 1993. Why idioms are not dead metaphors. *Idioms: Processing, structure, and interpretation*, pages 57–77.
Gibbs, Raymond W. 1994. *The poetics of mind: Figurative thought, language, and understanding*. Cambridge University Press.
Gibbs, Raymond W., Josephine M. Bogdanovich, Jeffrey R. Sykes, and Dale J. Barr. 1997. Metaphor in idiom comprehension. *Journal of memory and language*, 37(2):141–154.
Glucksberg, Sam, Matthew S. McGlone, and Deanna Manfredi. 1997. Property attribution in metaphor comprehension. *Journal of memory and language*, 36(1):50–67.
He, Xinran and Yan Liu. 2017. Not enough data?: Joint inferring multiple diffusion networks via network generation priors. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 465–474, Cambridge, UK, February 6-10. ACM.
Klyueva, Natalia, Antoine Doucet, and Milan Straka. 2017. Neural networks for multi-word expression detection. In *Proceedings of the 13th Workshop on Multiword Expressions*, page 60, Valencia, Spain, April 4.
Köper, Maximilian and Sabine Schulte im Walde. 2017. Improving verb metaphor detection by propagating abstractness to words, phrases and individual senses. In *Proceedings of the 1st Workshop on Sense, Concept and Entity Representations and their Applications*, pages 24–30, Valencia, Spain, April 4.
Krippendorff, Klaus. 2012. *Content analysis: An introduction to its methodology*. Sage.
Krčmář, Lubomír, Karel Ježek, and Pavel Pecina. 2013. Determining Compositionality of Expresssions Using Various Word Space Models and Measures. In *Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality*, pages 64–73, Sofia, Bulgaria, August 9.
Lakoff, George and Mark Johnson. 2008. *Metaphors we live by*. University of Chicago press.
Legrand, Joël and Ronan Collobert. 2016. Phrase representations for multiword expressions. In *Proceedings of the 12th Workshop on Multiword Expressions*, Berlin, Germany, August 11.
Lenci, Alessandro. 2008. Distributional semantics in linguistic and cognitive research. *Italian Journal of Linguistics*, 20(1):1–31.
Lenci, Alessandro. 2018. Distributional Models of Word Meaning. *Annual Review of Linguistics*, 4:151–171.
Levy, Omer, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225.
Lin, Dekang. 1999. Automatic identification of non-compositional phrases. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 317–324, College Park, USA, June 20-26.
Liu, Dilin. 2003. The most frequently used spoken american english idioms: A corpus analysis and its implications. *Tesol Quarterly*, 37(4):671–700.
McGlone, Matthew S. 1996. Conceptual metaphors and figurative language interpretation: Food for thought? *Journal of memory and language*, 35(4):544–565.
McGlone, Matthew S., Sam Glucksberg, and Cristina Cacciari. 1994. Semantic productivity and idiom comprehension. *Discourse Processes*, 17(2):167–190.
Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26tth International Conference on Neural Information Processing System*, pages 3111–3119, Stateline, USA, December 5-10.
Mikolov, Tomas, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics*, volume 13, pages 746–751, Altanta, USA, June 10-12.
Mitchell, Jeff and Mirella Lapata. 2010. Composition in Distributional Models of Semantics. *Cognitive Science*, 34(8):1388–1429.
Nunberg, Geoffrey, Ivan Sag, and Thomas Wasow. 1994. Idioms. *Language*, 70(3):491–538.
Pianta, Emanuele, Luisa Bentivogli, and Christian Girardi. 2002. MultiWordNet: Developing and Aligned Multilingual Database. In *Proceedings of the First International Conference on Global WordNet*, pages 293–302, Mysore, India, January 21-25.
Quartu, Monica B. 1993. *Dizionario dei modi di dire della lingua italiana*. RCS Libri.
Rei, Marek, Luana Bulat, Douwe Kiela, and Ekaterina Shutova. 2017. Grasping the finer point: A supervised similarity network for metaphor detection. *arXiv preprint arXiv:1709.00575*.

Rimell, Laura, Jean Maillard, Tamara Polajnar, and Stephen Clark. 2016. Relpron: A relative clause evaluation data set for compositional distributional semantics. *Computational Linguistics*, 42(4):661–701.

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 3rd International Conference on Computational Linguistics and Intelligent Text Processing*, pages 1–15, Mexico City, Mexico, February 17-23.

Senaldi, Marco S. G., Gianluca E. Lebani, and Alessandro Lenci. 2016. Lexical variability and compositionality: Investigating idiomaticity with distributional semantic models. In *Proceedings of the 12th Workshop on Multiword Expressions*, pages 21–31, Berlin, Germany, August 11.

Senaldi, Marco S. G., Gianluca E Lebani, and Alessandro Lenci. 2017. Determining the compositionality of noun-adjective pairs with lexical variants and distributional semantics. *Italian Journal of Computational Linguistics*, 3(1):43–58.

Steen, Gerard J., Aletta G. Dorst, J. Berenike Herrmann, Anna Kaal, Tina Krennmayr, and Trijntje Pasma. 2014. A method for linguistic metaphor identification: From mip to mipvu. *Metaphor and the Social World*, 4(1):138–146.

Tanguy, Ludovic, Franck Sajous, Basilio Calderone, and Nabil Hathout. 2012. Authorship attribution: Using rich linguistic features when training data is scarce. In *PAN Lab at CLEF*, Valencia, Spain, September 23-26.

Titone, Debra and Maya Libben. 2014. Time-dependent effects of decomposability, familiarity and literal plausibility on idiom priming: A cross-modal priming investigation. *The Mental Lexicon*, 9(3):473–496.

Torre, Enrico. 2014. *The emergent patterns of Italian idioms: A dynamic-systems approach*. Ph.D. thesis, Lancaster University.

Turney, Peter D. and Patrick Pantel. 2010. From Frequency to Meaning: Vector Space Models of Semantics. *Journal of Artificial Intelligence Research*, 37:141–188.

Wulff, Stefanie. 2008. *Rethinking Idiomaticity: A Usage-based Approach*. Continuum.

Deep Learning for Automatic Image Captioning in Poor Training Conditions

Caterina Masotti*
Università di Roma, Tor Vergata

Danilo Croce**
Università di Roma, Tor Vergata

Roberto Basili†
Università di Roma, Tor Vergata

Recent advancements in Deep Learning have proved that an architecture that combines Convolutional Neural Networks and Recurrent Neural Networks enables the definition of very effective methods for the automatic captioning of images. The disadvantage that comes with this straightforward result is that this approach requires the existence of large-scale corpora, which are not available for many languages.

This paper introduces a simple methodology to automatically acquire a large-scale corpus of 600 thousand image/sentences pairs in Italian. At the best of our knowledge, this corpus has been used to train one of the first neural captioning systems for the same language. The experimental evaluation over a subset of validated image/captions pairs suggests that the achieved results are comparable with the English counterpart, despite a reduced amount of training examples.

1. Introduction

The image captioning task consists of generating a brief description in natural language of a given image that is able to capture the depicted objects and the relations between them, as discussed in (Bernardi et al. 2016). More precisely, given an image I as input, an *image captioner* should be able to generate a well-formed sentence $S(I) = (s_1, ..., s_m)$, where every s_i is a word from a vocabulary $V = \{w_1, ..., w_n\}$ in a given natural language. Some examples of images and corresponding captions are reported in Figure 1. This task is rather complex as it involves non-trivial subtasks to solve, such as object detection, mapping visual features to text and generating text sequences.

Recently, neural methods based on deep neural networks have reached impressive state-of-the-art results in solving this task (Karpathy and Li 2015; Mao et al. 2014; Xu et al. 2015). One of the most successful architectures implements the so-called *encoder-decoder* end-to-end structure (Goldberg 2016).
Differently by most of the existing encoder-decoder structures, in (Vinyals et al. 2014) the encoding of the input image is performed by a convolutional neural network which transforms it in a dense feature vector; then, this vector is "translated" to a descriptive sentence by a Long Short-Term Memory (LSTM) architecture, which takes the vector

* Dept. of Enterprise Engineering - Via del Politecnico 1, 00133 Rome, Italy.
E-mail: caterinamasotti@yahoo.it
** Dept. of Enterprise Engineering - Via del Politecnico 1, 00133 Rome, Italy.
E-mail: croce@info.uniroma2.it
† Dept. of Enterprise Engineering - Via del Politecnico 1, 00133 Rome, Italy.
E-mail: basili@info.uniroma2.it

(a) English: *A yellow school bus parked in a handicap spot*, Italian: *Uno scuolabus giallo parcheggiato in un posto per disabili.*

(b) English: *A cowboy rides a bucking horse at a rodeo*, Italian: *Un cowboy cavalca un cavallo da corsa a un rodeo.*

(c) English: *The workers are trying to pry up the damaged traffic light*, Italian: *I lavoratori stanno cercando di tirare su il semaforo danneggiato.*

Figure 1
Three images from the MSCOCO dataset, along with two human-validated descriptions, one for English and one for Italian.

as the first input and generates a textual sequence starting from it. This neural model is very effective, but also very expensive to train in terms of time and hardware resources[1], because there are many parameters to be learned; not to mention that the model is overfitting-prone, thus it needs to be trained on a training set of annotated images that is as large and heterogeneous as possible, in order to achieve a good generalization capability.

Hardware and time constraints do not always allow to train a model in an optimal setting, and, for example, cutting down on the dataset size could be necessary: in this case we have *poor training conditions*. Of course, this reduces the model's ability to generalize on new images at captioning time.
Another cause of poor training conditions is the lack of a good quality dataset, for example in terms of annotations: the manual captioning of large collections of images requires a lot of effort and, as of now, human-annotated datasets only exist for a restricted set of languages, such as English. As a consequence, training such a neural model to produce captions in another language (e.g. in Italian) is an interesting problem to explore, but also challenging due to the lack of data resources.

A viable approach is building a resource by *automatically translating the annotations from an existing dataset*: this is much less expensive than manually annotating images, but of course it leads to a loss of human-like quality in the language model. This approach has been adopted in this work to perform one of the first neural-based image captioning in Italian: more precisely, the annotations of the images from the MSCOCO dataset, one of the largest datasets in English of image/caption pairs, have been automatically translated to Italian in order to obtain a first resource for this language. This has been exploited to train a neural captioner, whose quality can be improved over time (e.g., by manually validating the translations). Then, a subset of this Italian dataset has been used as training data for the neural captioning system defined in (Vinyals et al. 2014), while a subset of the test set has been manually validated for evaluation purposes.

1 As of now, training a neural encoder-decoder model such as the one presented at `http://github.com/tensorflow/models/tree/master/im2txt` on a dataset of over $580,000$ image-caption examples takes about two weeks even with a very performing GPU.

In particular, prior to the experimentations in Italian, some early experiments have been performed with the same training data originally annotated in English, to get a reference benchmark about convergence time and evaluation metrics on a dataset of smaller size. These results in English will suggest if the Italian image captioner shows similar performance when trained over a reduced set of examples or the noise induced in the automatic translation process compromises the neural training phase. Moreover, these experiments have also been performed with the introduction of a pre-trained word embedding (derived using the method presented in (Mikolov et al. 2013)), in order to measure how it affects the quality of the language model learned by the captioner, with respect to a randomly initialized word embedding that is learned together with the other model parameters.

Overall the contributions of this work are threefold: (*i*) the investigation of a simple, automatized way to acquire (possibly noisy) large-scale corpora for the training of neural image captioning methods in poor training conditions; (*ii*) the manual validation of a first set of human-annotated resources in Italian; (*iii*) the implementation of one of the first automatic neural-based Italian image captioners.

In the rest of the paper, the adopted neural architecture is outlined in Section 2. The description of a brand new resource for Italian is presented in Section 3. Section 4 reports the results of the early preparatory experimentations for the English language and then the ones for Italian. Finally, Section 5 derives the conclusions.

2. The Show and Tell Architecture

The Deep Architecture considered in this paper is the *Show and Tell* architecture, described in (Vinyals et al. 2014) and sketched in Figure 2. It follows an encoder-decoder structure where the image is encoded in a dense vector by a state-of-the-art deep CNN, in this case *InceptionV3* (Szegedy et al. 2016), followed by a fully connected layer; the resulting feature vector is fed to a LSTM, used to generate a text sequence, i.e. the caption.

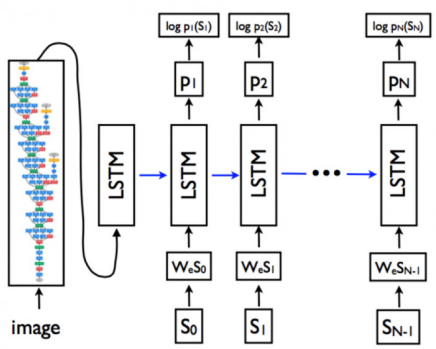

Figure 2
The Deep Architecture presented in (Vinyals et al. 2014). LSTM model combined with a CNN image embedder and word embeddings. The unrolled connections between the LSTM memories are in blue.

As the CNN encoder has been trained over an object recognition task, it allows encoding the image in a dense vector that is strictly connected to the entities observed in the image.

At the same time, the LSTM implements a language model, in line with the idea introduced in (Mikolov et al. 2010): it captures the probability of generating a given word in a string, given the words generated so far. In the overall training process, the main objective is to train a LSTM to generate the next word given not only the string produced so far, but also a set of image features.

Since the CNN encoder is (mostly) language independent, due to the fact that it is trained to recognize visual features that are not related to natural language, it can be totally re-used even in the captioning of images in other languages, such as Italian. On the contrary, the language model underlying the LSTM needs new examples to be trained.

In the following subsections, further details will be provided on the CNN and LSTM components of the *Show and Tell* architecture.

2.1 CNN image encoder: InceptionV3

As said before, the image-encoding CNN is the *InceptionV3* network, whose details can be found in (Szegedy et al. 2016): it is a modified version of GoogLeNet introduced in (Szegedy et al. 2015), whose architecture consists in repeated *Inception* modules: they are a combination of *convolutional layers*, that analyze adjacent groups of features coming as output from the previous Inception modules in the network, and *pooling layers*, which output a function f (in this case, max) of incoming inputs from the convolutional phase. Their structure is shown in Figure 3 and Figure 4.

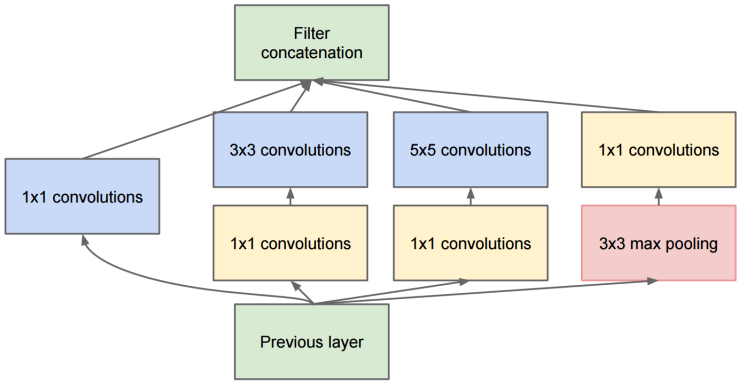

Figure 3
The original Inception module in GoogLenet.

InceptionV3 has shown state-of-the-art results in various visual recognition tasks: before being integrated in this architecture its weights are pre-trained on the image classification task.

The last layer of the network is a fully connected one and outputs a fixed-length vector representation of the image: this vector will be the first input fed to the LSTM.

2.2 LSTM decoder

The LSTM is the part of the architecture that takes as input the visual feature vector that comes as output from the CNN, and translates it to a descriptive sentence. At every time step, it outputs a softmax vector of probabilities over all the words in the vocabulary:

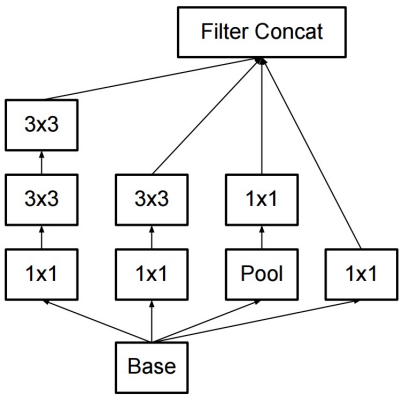

Figure 4
The new Inception module after refactoring 5×5 convolutions.

its elements represent the conditional probabilities $P(S_t|I, S_0, ..., S_{t-1})$ of the t-th word of the sequence. The input at the first time step is the output of the CNN, then it is the word vector $\mathbf{v}_{s_t}$ of the last word predicted s_t. The input, forget and output layers are sigmoidals functions: they are used as filters in order to choose the relevant parts of the sequence generated until that moment. All the definitions can be found below, where the notation and the figure are the same used in the papers (Vinyals et al. 2014) and (Vinyals et al. 2017):

$$
\begin{aligned}
i_t &= \sigma(W_{ix}x_t + W_{im}m_{t-1}) \\
f_t &= \sigma(W_{fx}x_t + W_{fm}m_{t-1}) \\
o_t &= \sigma(W_{ox}x_t + W_{om}m_{t-1}) \\
c_t &= f_t \odot c_{t-1} + i_t \odot h(W_{cx}x_t + W_{cm}m_{t-1}) \\
m_t &= o_t \odot c_t \\
p_{t+1} &= Softmax(m_t)
\end{aligned}
$$

As in (Vinyals et al. 2014), the core of the LSTM model is a memory cell *c* encoding knowledge at every time step of what inputs have been observed up to this step (as reported in Figure 5. The cell is controlled by *gates* ,Äì layers which are applied multiplicatively and thus can either keep a value from the gated layer if the gate is 1 or zero such value if the gate is equal to 0. In particular, three gates are being used which control whether to forget the current cell value (forget gate *f*): the memory block contains a cell *c* which is read its input (input gate *i*) and whether to output the new cell value (output gate *o*). Here $\odot$ represents the product with a gate value, and the various W matrices are trained parameters. The nonlinearities are sigmoid $\sigma(\cdot)$ and hyperbolic tangent $h(\cdot)$. Finally m_t is used to feed to a $Softmax$, which will produce a probability distribution p_t over all words.

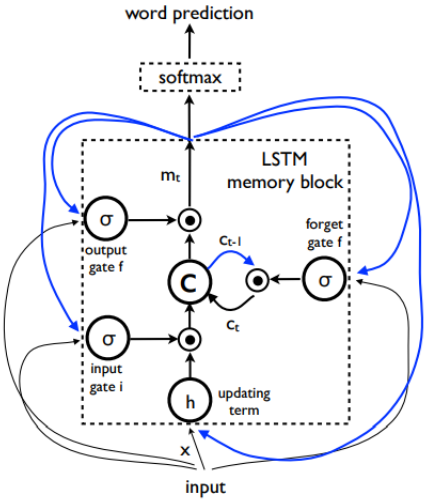

Figure 5
The gates and connections of the LSTM.

At training time, the loss function is the sum of the negative log-likelihoods for the generated words at every time step t:

$$L(I, S) = -\sum_{t=1}^{N} \log P_t(S_t). \quad (1)$$

The loss is minimized with respect to the LSTM parameters, the weights of the last fully connected layer of the CNN and the weights of the embedding matrix W.

At inference time, the used strategy to develop the best sequence starting from the output probability distribution is called **beam search**: at every time step t the k sentences with highest probabilities are kept and their $t+1$-th word is predicted. Finally, among the $k \times k$ generated sentences of length $t+1$, only the k best are chosen and expanded in the next step, and so on. In (Vinyals et al. 2017), the best results have been achieved with beam size $k = 3$, and this parameter has been kept in the experimental setting.

In this work, we will train this architecture over a corpus that has been automatically translated from the MSCOCO dataset. We thus speculate that the LSTM will learn a sort of simplified language model, more inherent to the automatic translator than to an Italian speaker. However, we are also convinced that the quality achievable by modern translation systems (Luong, Pham, and Manning 2015), combined with the generalization that can be obtained by a LSTM trained over thousands of (possibly noisy) translations will be able to generate reasonable and intelligible captions.

3. Automatic acquisition of a Corpus of Captions in Italian

In this section we present the first release of the MSCOCO-it, a new resource for the training of data-driven image captioning systems in Italian. It has been built starting from the MSCOCO dataset for English (Lin et al. 2014): in particular we considered the training and validation subsets, made respectively of $82,783$ and $40,504$ images, where every image has 5 human-written annotations in English.

The Italian version of the dataset has been acquired with an approach that automatizes the translation task: for each image, all its five annotations have been translated by using an automatic translator[2]. The result is a big amount of visual data annotated with multiple sentences: the annotations in English are fully translated to Italian, but not of the best quality with respect to the Italian fluent language. This automatically translated data can be used to train a model, but for the evaluation a test set of human-validated examples is needed: so, the translations of a subset of the MSCOCO-it have been manually validated.

In (Vinyals et al. 2014), two subsets of $2,024$ and $4,051$ images from the MSCOCO validation set have been held out from the rest of the data and have been used for development and testing of the model, respectively. A subset of these images has been manually validated: 308 images from the development set and 596 from the test set. In Table 1, statistics about this brand new corpus are reported, where the specific amount of unvalidated (*u.*) and validated (*v.*) data is made explicit[3].

Table 1
Statistics about the MSCOCO-it corpus. *p.* stands for *partially validated*, since some images have only some validated captions out of five. The partially validated images are between parentheses because they are already counted in the validated ones.

		#images	#captions	#words
training	*u.*	116,195	581,286	6,900,546
development	*v.*	308	1,516	17,913
	u.	1,696	8,486	101,448
	p.	(14)	25	304
test	*v.*	596	2,941	34,657
	u.	3,422	17,120	202,533
	p.	(23)	41	479
total		**122,217**	**611,415**	**7,257,880**

4. Experimental Evaluation

In order to be consistent with a scenario characterized by *poor training conditions* (limited hardware resources and time constraints) all the experimentations in this paper have been made by training the model on significantly smaller samples of data with respect to the whole MSCOCO dataset (made of more than $583,000$ image-caption examples).

First of all, some early experimentations have been performed on smaller samples of data from MSCOCO in English, in order to measure the loss of performance caused by the reduced size of the training set[4]. Each training example is a image-caption pair and these have been grouped in data *shards* during the training phase: each shard contains about $2,300$ image-caption examples. The model has been trained on datasets of $23,000$, $34,500$ and $46,000$ image-caption pairs (less than 10% of the entire dataset). In order to

2 Sentences have been translated by using Bing Translator (`https://www.bing.com/translator`) between December 2016 and January 2017.
3 Although Italian annotations are available for all the images of the original dataset, in the table some images were not counted because they are corrupted and therefore have not been used.
4 A proper tuning phase was too expensive so we adopted the parameters provided in `https://github.com/tensorflow/models/tree/master/im2txt`

balance the reduced size of the training material and provide some kind of linguistic generalization, we evaluated the adoption of pre-trained word embedding in the training/tagging process. In fact, in (Vinyals et al. 2014) the LSTM architecture initializes randomly all vectors representing input words; these are later trained together with the other parameters of the network.

We wondered if a word embedding already pre-trained on a large corpus could help the model to generalize better on brand new images at test time. The neural model chosen for this experiment, in order to learn a word embedding which could effectively represent a proper natural language model, is the Skip-gram model. The Skip-gram model is one of the variations of the *word2vec* language model, introduced by Mikolov et al. in (Mikolov et al. 2013) and (Mikolov et al. 2013) in 2013, which consists in training a shallow feed-forward network in order to learn word embedding vectors: its structure is made of an embedding layer and an output layer. The embedding layer maps the input to d-embedding vectors accordingly to the weights matrix C, while the output computes a probability distribution over words. *word2vec*'s success comes from being able to produce word vectors that, linearly combined, catch meaningful semantic properties between words. The Skip-gram model, applied to *word2vec*, given a word w_i, predicts its context words (the words that could possibly surround w_i) $\{w_{1,i}, ..., w_{C,i}\}$.

The input word w_i is mapped to a vector in the weights matrix W of the embedding layer and then a vector h is computed from it; the output are C probability distribution softmax vectors for the context words $\{w_{1,i}, ..., w_{C,i}\}$. The architecture of the Skip-gram neural model is reported in Figure 6.

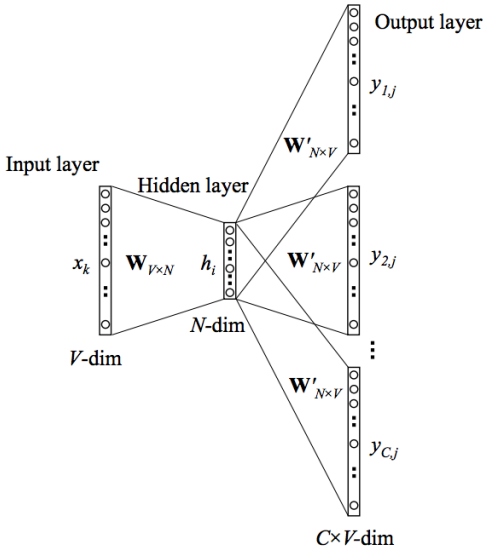

Figure 6
The Skip-gram model.

Due to this embedding model's ability to capture many different semantic shades of the natural language's words, we speculate that it could help the Show and Tell captioner when the amount of training data is too small to generalize, and therefore introduce in the model a word embedding learned through a Skip-gram model from an English dump of Wikipedia. The LSTM architecture has been trained on the same shards but initializing the word vectors with this pre-trained word embedding.

Table 2
Results on `im2txt` for the English language with a training set of reduced size, without / with the use of a pre-trained word embedding. Moreover benchmark results are reported.

# Shards	BLEU-4	METEOR	CIDEr
1	10,1 / 11,5	13,4 / 13,1	18,8 / 24,4
2	15,7 / 18,9	18,2 / 16,3	36,1 / 51,9
5	22,0 / 22,7	20,2 / 20,4	64,1 / 65,0
10	22,4 / 24,7	22,0 / 21,7	73,2 / 73,7
20	26,5 / 26,2	21,9 / 22,3	79,3 / 79,1
NIC (Vinyals et al. 2014)	27,7	23,7	85,5
NICv2 (Vinyals et al. 2014)	32,1	25,7	99,8
`im2txt`	31,2	25,5	98,1

Table 2 reports results on the English dataset in terms of BLEU-4, CIDEr and METEOR, the same used in (Vinyals et al. 2014): they are commonly used in machine translation evaluation to evaluate the quality of an automatic translation in comparison to other human-written reference sentences. Since the sentences generated by *Show and Tell* are *"translations"* of an image, it makes sense to apply these metrics to evaluate how good the generated captions are. Some details about them:

- **BLEU-4:** a geometric mean of the modified n-gram precisions on the whole corpus, penalized by a *brevity penalty* for too short sentences.
- **CIDEr**: an average, on n-grams, of average cosine similarities between the sentences to evaluate and their human-validated reference sentences.
- **METEOR**: a metric which creates a *matching* between unigrams of the sentence to evaluate and the validated reference sentence. The score is computed by combining the unigram precision, the unigram recall and a fragmentation penalty.

In the first five rows of Table 2, results are reported both in the case of randomly initialized word embedding and pre-trained ones. We compare these results with the ones achieved by the original NIC and NICv2 networks presented in (Vinyals et al. 2014), and the ones measured by testing a model available in the web[5], trained on the original whole training set (the last row, referred as `im2txt`).

Results obtained by the network when trained on a reduced dataset are clearly lower w.r.t. the NIC results, but it is straightforward that similar result are obtained, especially considering the reduced size of the training material. The contribution of pre-trained word embeddings is not significant, in line with the findings from (Vinyals et al. 2014). However, it is still interesting noting that the lexical generalization of this unsupervised word embeddings is beneficial, especially when the size of the training material is minimal (e.g. when one shard is used, especially if considering the CIDEr metrics). As the amount of training data grows, its impact on the model decreases, until it is not significant anymore.

5 `http://github.com/tensorflow/models/issues/466`

Table 3
Metrics for the experimentations on `im2txt` for the Italian language with a training set of reduced size, without / with and the use of a pre-trained word embedding.

# Shards	BLEU-4	METEOR	CIDEr
1	11.7 / 12.9	16.4 / 16.9	27.4 / 29.4
2	16.9 / 17.1	18.8 / 18.7	45.7 / 45.6
5	22.0 / 21.4	21.2 / 20.9	62.5 / 60.8
10	22.4 / 22.9	22.0 / 21.5	71.9 / 68.8
20	23.7 / 23.8	22.2 / 22.0	73.0 / 73.2

For what concerns the results on Italian, the experiments have been performed by training the model on samples of $23,000$, $34,500$ and $46,000$ examples that are the counterpart of the ones used for English, where the captions are automatically translated with Bing. The model has been evaluated against the validated sentences, and results are reported in Table 3. Results are impressive as they are in line with the English counterpart. It supports the robustness of the adopted architecture, as it seems to learn even from a noisy dataset of automatically translated material. Most importantly, it confirms the applicability of the proposed simple methodology for the acquisition of datasets for image captioning.

When trained with 20 shards, the Italian captioner generates the following description of the images shown in Figure 1: Image 1a: "*Un autobus a due piani guida lungo una strada.*", Image 1b: "*Un uomo che cavalca una carrozza trainata da cavalli.*", Image 1c: "*Una persona che cammina lungo una strada con un segnale di stop.*"

An attempt to use a word embedding that has been pre-trained on a large corpus (more precisely, on a dump of Wikipedia in Italian) has also been made, but the empirical results reported in Table 3 show that its contribution is not relevant but still significant when fewer examples are adopted. This confirm the beneficial impact of word embedding in neural training when the size of the labeled material is reduced, while it seems neglected when this amount grows.

4.1 Automatic translation: before or after sentence generation?

After demonstrating the viability of the proposed approach, we argued the following research question: *is it necessary to translate the sentences of the MSCOCO training set in order to let the model learn a more accurate Italian, or does the translation of the sentences generated from a model trained on English provide the same language quality?*

In order to answer to such question, the validated portion of the MSCOCO-it test set has been captioned by the `im2txt` model (see Table 2) trained on the *whole* English dataset (about $580,000$ training examples). Then, the generated captions have been automatically translated to Italian. Finally, these captions have been compared with the ones produced by the best model trained on Italian (training set of $46,000$ examples, see Table 3), in terms of quality, by comparing their BLEU-4, METEOR and CIDER metrics. The results are reported in the table below.

Results shown in table 4 suggest that training a model on a dataset already translated in Italian (last row) seems to achieve a better sentence quality w.r.t. performing the automatic translation task after the captions have already been generated from a model in English (row referred as `im2txt` + automatic translation: it is worth noting

Table 4
Results for the Italian language, by translating the training set and feeding it to the model and by translating the captions after being generated from the `im2txt` English model.

# Model	BLEU-4	METEOR	CIDEr
`im2txt` + automatic translation	21,6	21,8	70,9
Best model trained on Italian	23,8	22,0	73,2

(a) `im2txt`+translation: *Un giocatore di baseball che oscilla una mazza ad una sfera*, Italian model: *Un giocatore di baseball che tiene una mazza da baseball su un campo.*

(b) `im2txt`+translation: *Una grande torre dell'orologio che sovrasta una cittá*, Italian model: *Un grande edificio con un orologio sulla parte superiore.*

(c) `im2txt`+translation: *Un gruppo di persone che cavalcano sulle spalle dei cavalli*, Italian model: *un uomo che cavalca un cavallo in un campo.*

(d) `im2txt`+translation: *Una persona che salta una tavola skate in aria*, Italian model: *Un uomo che cavalca uno skateboard su una strada.*

Figure 7
Some images from the MSCOCO dataset, along with their descriptions generated from both the model trained on the full English dataset, with a subsequent translation to Italian, and the model trained directly on Italian.

the advantage of the model trained directly on Italian, although the number of training examples is significantly smaller than the other model ($580,000$ examples in English vs $46,000$ in automatically translated Italian).

We report a comparative analysis of some of the sentences generated from both models. Some images from the MSCOCO test set, along with their captions produced by both models, are reported in Figure set 7. Some of the differences that emerge from the captions are that the sentenced obtained by translating the caption of the English

`im2txt` model better captures *actions*, probably due to seeing many more training examples, but this ability is then heavily penalized by the noise introduced by the automatic translation (often raw and literal) as we can see in Figure 7a: the action of moving the baseball bat towards the ball (*[...] oscilla una mazza a una sfera*) has been described in the English sentence, but has been translated poorly in Italian. Meanwhile, the caption produced from the model trained on automatically translated Italian, even if the *swinging* action has not been captured, produces a less exhaustive but acceptable (from a linguistic perspective) description.

A similar phenomenon can be observed in Figure 7b: a verb is inserted in the English caption by `im2txt`, but even if this verb is properly translated, the automatic sentence translation is less accurate somewhere else (*torre dell'orologio*), while the Italian model produces a simpler but clearer description.

In Figure 7c, the model trained on Italian surprisingly captures more accurately the visual features (*Un uomo che cavalca*, a single person, not more than one), while the other model confuses the walking action of the group of persons ahead with the act of riding horses, probably due to their proximity: once more, the English model tries to provide more details about the action of riding horses (*[...]che cavalcano sulle spalle dei cavalli*), but the automatic translation makes it sound a bit rough.

What actually emerges is that the model trained directly on a dataset in Italian tends to learn simpler descriptions, with few action-describing verbs, that are clearer to read and less error-prone; the model trained on the whole English dataset has certainly seen more examples and is able to capture more actions from a scene, but the more syntactically complex are the sentences, the more error-prone they become when translated automatically.

5. Conclusions

In this paper a simple methodology for the training of neural models for the automatic captioning of images is presented, along with a large-scale dataset of about $600,000$ image captions in Italian produced by using an automatic machine translator. Although the noise introduced in this step, it allows to train one of the first neural-based image captioning systems for Italian.

The quality of this system seems comparable with the English counterpart, if trained over a comparable set of data: these results are impressive and confirm the robustness of the adopted neural architecture. Moreover, empirical evidence tell us that the approach of training a system on an Italian translated dataset produces sentences with a higher accuracy, with respect to producing captions with an English system, even if trained on many more visual and language features, and then translating them automatically.

We believe that the obtained resource paves the way to the definition and evaluation of Neural Models for Image captioning in Italian, and we hope to contribute to the Italian Community, hopefully using the validated dataset in a future Evalita[6] campaign.

References

Bernardi, Raffaella, Ruket Cakici, Desmond Elliott, Aykut Erdem, Erkut Erdem, Nazli Ikizler-Cinbis, Frank Keller, Adrian Muscat, and Barbara Plank. 2016. Automatic description generation from images: A survey of models, datasets, and evaluation measures. *Journal of Artificial Intelligence Research*, 55(1):409–442, January.

6 `http://www.evalita.it/`

Goldberg, Yoav. 2016. A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research*, 57(1):345–420, September.

Karpathy, Andrej and Fei-Fei Li. 2015. Deep visual-semantic alignments for generating image descriptions. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015*, pages 3128–3137, Boston, MA, USA, June 7-12. IEEE Computer Society.

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 13th European Conference*, pages 740–755, Zürich, Switzerland, September 6-12. Springer International Publishing.

Luong, Thang, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015*, pages 1412–1421, Lisbon, Portugal, September 17-21. The Association for Computational Linguistics.

Mao, Junhua, Wei Xu, Yi Yang, Jiang Wang, and Alan L. Yuille. 2014. Deep captioning with multimodal recurrent neural networks (m-rnn). *CoRR*, abs/1412.6632.

Mikolov, Tomas, Martin Karafiát, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association*, pages 1045–1048, Makuhari, Chiba, Japan, September 26-30, 2010.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In Christopher J. C. Burges, Leon Bottou, Max Welling, Zoubin Ghahramani, and Kilian Q. Weinberger, editors, *Proceedings of the 26th International Conference on Neural Information Processing Systems NIPS'13*. Curran Associates, Inc., Lake Tahoe, Nevada (USA), December 05 - 10, 2013, pages 3111–3119.

Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015*, pages 1–9, June 7-12.

Szegedy, Christian, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, pages 2818–2826, Las Vegas, NV, USA, June 27-30.

Vinyals, Oriol, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2014. Show and tell: A neural image caption generator. *CoRR*, abs/1411.4555.

Vinyals, Oriol, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2017. Show and tell: Lessons learned from the 2015 MSCOCO image captioning challenge. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):652–663.

Xu, Kelvin, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2048–2057, Lille, France, 07–09 Jul. PMLR.

Deep Learning of Inflection and the Cell-Filling Problem

Franco Alberto Cardillo*
ILC-CNR

Marcello Ferro*
ILC-CNR

Claudia Marzi*
ILC-CNR

Vito Pirrelli*
ILC-CNR

Machine learning offers two basic strategies for morphology induction: lexical segmentation and surface word relation. The first approach assumes that words can be segmented into morphemes. Inferring a novel inflected form requires identification of morphemic constituents and a strategy for their recombination. The second approach dispenses with segmentation: lexical representations form part of a network of associatively related inflected forms. Production of a novel form consists in filling in one empty node in the network. Here, we present the results of a task of word inflection by a recurrent LSTM network that learns to fill in paradigm cells of incomplete verb paradigms. Although the task does not require morpheme segmentation, we show that accuracy in carrying out the inflection task is a function of the model's sensitivity to paradigm distribution and morphological structure.

1. Introduction

Following a morpheme-based tradition in morphological inquiry, the process of morphology induction can be defined as the task of singling out morphological formatives from fully inflected word forms. Formatives are understood to be part of the morphological lexicon, where they are accessed for word recognition, and retrieved and spelled out for word production. The view requires that a word form be segmented into meaningful sublexical signs, called *morphemes*, each contributing a separable piece of morpho-lexical content. In inflecting languages, typically this holds for regularly inflected forms, as with Italian *cred-ut-o* 'believed' (past participle, from CREDERE 'believe'), where *cred-* conveys the lexical meaning of CREDERE, and *-ut-o* is associated with morpho-syntactic features. A further assumption is that there always exists an underlying *base form* from which all other forms are spelled out. In an irregular verb form like Italian *appes-o* 'hung' (from APPENDERE), however, it soon becomes difficult to separate morpho-lexical information (the verb stem) from morpho-syntactic information (e.g. past participle). Various technical attempts have been made to circumvent these problems and restore the disrupted biuniqueness between forms and morpho-sysntactic content in non trivial inflectional systems. In fact, for too many languages, morpheme segmentation is an ill-defined task, due to notorious problems with the classical, sign-based notion of morpheme, and the non-segmental processes of introflexive (i.e. root and pattern), tonal and apophony-based morphologies. So, the assumption that

* Istituto di Linguistica Computazionale "A. Zampolli" - via G. Moruzzi 1, 56124 Pisa, Italy.
E-mail: name.surname@ilc.cnr.it

any word form can uniquely and consistently be segmented into morphemic sublexical constituents is at best dubious, and cannot be entertained as a general bootstrapping hypothesis for morphology learning.

A different formulation of the same task assumes that the lexicon consists of fully-inflected word forms and that morphology induction is the result of discovering implicative relations between them. Unknown forms are inferred through redundant analogy-based patterns between known forms, along the lines of an analogical proportion such as:

rendere 'make' :: *reso* 'made' = *appendere* 'hang' :: *appeso* 'hung'.

Support to this view comes from developmental psychology, where words are understood as the foundational elements of language acquisition, from which early grammar rules emerge epiphenomenally (Tomasello 2000; Goldberg 2003). After all, children are exposed to fully inflected forms in acquisition, and have no privileged access to underlying base forms. Besides, they appear to be extremely sensitive to sub-regularities holding between inflectionally-related forms (Bittner, Dressler, and Kilani-Schoch 2003; Colombo et al. 2004; Dąbrowska 2004; Orsolini and Marslen-Wilson 1997; Orsolini, Fanari, and Bowles 1998). Further support is lent by neurobiologically inspired computer models of language, blurring the traditional dichotomy between processing and storage (Elman 2009; Marzi et al. 2016).

In the present paper, we will mainly be concerned with issues of cognitive plausibility and inter-linguistic coverage for computational models of word generation. Our main emphasis here is not on the most effective machine learning strategy for a specific language classification task on a specific language, but on the general algorithmic requirements of the developmentally realistic task we set ourselves to (namely the 'cell-filling problem', see infra). In particular, we will focus on how these requirements are met by one of the most advanced and sophisticated models of recurrent neural networks to date, so-called Long Short Term Memory (LSTM) networks (Bengio, Simard, and Frasconi 1994; Hochreiter and Schmidhuber 1997; Malouf 2017). Hence, comparison of the system performance with other competing systems will be carried out only to the extent needed to show that the proposed architecture compares reasonably well with state of the art algorithms and to focus on its learning bias. As we are not interested in proving that our LSTM architecture performs better than other systems, but only in assessing its potential with the sparsest possible set of language-specific assumptions, issues of task-driven, language-driven and parameter-driven optimization are not addressed.

2. The cell-filling problem

To understand how word inflection can be conceptualised as a word relation task, it is useful to think of this task as a *cell-filling problem* (Ackerman and Malouf 2013; Ackerman, Blevins, and Malouf 2009). Inflected forms are traditionally arranged in so-called *paradigms*. The full paradigm of CREDERE 'believe' is a labelled set of all its inflected forms: *credere, credendo, creduto, credo* etc. In most cases, these forms take one and only one *cell*, defined as a specific combination of tense, mood, person and number features: e.g. *crede*, PRES IND, 3S. In all languages, words happen to follow a Zipfian distribution, with very few high-frequency words, and a very long tail of exceedingly rare words (Blevins, Milin, and Ramscar 2017). As a result, even high-frequency paradigms happen to be attested partially, and speakers must then be able to generalise incomplete paradigmatic knowledge. This amounts to a cell-filling problem:

given a set of attested forms in a paradigm, the speaker has to guess what other forms can fill in empty cells in the same paradigm.

2.1 Computational modelling

Borrowing Blevins' (2006) terminology, we can make a distinction between "constructive" and "abstractive" algorithms for word learning. Constructive algorithms assume that classificatory information is morpheme-based. Word forms are segmented into morphemes for training, and a classifier must learn to apply morpheme segmentation to novel forms after training. An abstractive learning algorithm, on the other hand, sees morphological structure as emerging from full forms, be they annotated with classificatory information (supervised mode) or not (unsupervised mode). From this perspective, training data consist of unsegmented word forms (strings of either letters or sounds), possibly coupled with their lexical and morpho-syntactic content. Accordingly, morphological acquisition boils down to learning from lexical representations in training, to generalise them to unknown forms. In this process, word-internal constituents can possibly emerge, either as a result of the formal redundancy of raw input data (unsupervised mode), or as a by-product of form-content mappings (supervised mode). Only abstractive machine learning models of inflection can address the cell-filling problem. Thus, we will hereafter focus on abstractive models.

A further important qualification to be made in this connection concerns the set of a-priori assumptions about the target inflection system that some abstractive algorithms avail themselves of. For example, knowledge that the target language morphology is concatenative can considerably constrain the hypothesis search space of the algorithm, which is biased to look for stem-ending patterns only. This bias has important implications for word acquisition. Although no explicit morpheme segmentation is provided in training, the way word forms are tentatively split into internal constituents brings to bear detailed information about boundary relations between constituents (Goldsmith 2001). Other a-priori biases may consist in (i) using fixed-length positional templates (Keuleers and Daelemans 2007; Plunkett and Juola 1999), or (ii) tying individual symbols (letters or sounds) to specific positions in the input representation (so-called "conjunctive" coding) (Coltheart et al. 2001; Harm and Seidenberg 1999; McClelland and Rumelhart 1981; Perry, Ziegler, and Zorzi 2007; Plaut et al. 1996), or (iii) resorting to some language-specific alignment algorithms (Albright 2002) or head-and-tail splitting procedures (Pirrelli and Yvon 1999).

We contend that a bootstrapping algorithm for morphology induction should be valued for its ability to converge on the acquisition of an inflection system with the sparsest possible set of a-priori assumptions about the underlying structure of the system, rather than for its learning bias. The ability to recognise position-independent patterns in symbolic time series, like the word *book* in *handbook*, or the verb root *mach* in German *gemacht* ('made' past participle), lies at the heart of human learning of inflection. A more human-like algorithm for morphological bootstrapping should have the capacity to adapt itself to the morphological structure of the target language. This is all the more important, if we consider that the way morpho-syntactic features are contextually realised through processes of word inflection arguably represents the widest dimension of crosslinguistic grammatical variation (somewhat belittling universal invariances along other dimensions (Evans and Levinson 2009)). Although many comprehensive catalogues of the morphological markers and patterns in a given language or languages are available (Bickel and Nichols 2005; McWorther 2001; Shosted 2006), there exists no close inventory of parametric cross-linguistic variation for inflection.

In the machine learning literature, a more principled approach to morphology induction has been taken by so-called "features and classes" approaches (McNamee, Nicholas, and Mayfield 2009; Pauw and Wagacha 2007), where a word form is represented as a set of redundantly specified n-grams, i.e. possibly overlapping substrings of *n* characters making up the input string: for example, 'wa', 'al', and 'lk' for the string *walk*. N-grams have no internal structure and may be order-independent. The algorithm may start with the hypothesis that each word form is in a class of its own, and uses a stochastic classifier to calculate the conditional probability of having a certain class (a word form) given the set of distributed n-grams associated with the class. N-grams that occur in many words will be poorly discriminative, whereas features that happen to be repeatedly associated with a few word forms only will be given a morphologically meaningful interpretation. Features and classes approaches are in a position to address the bootstrapping issue of converging on the appropriate morphological classification of input data with no a-priori learning biases. However, due to their n-gram bases representations of input forms, it is not clear how they can be used for generating a fully spelled out form from its lexical and morpho-syntactic features, as required by the cell-filling problem.

A few more recent connectionist models have addressed the problem of learning inflectional paradigms. Goldsmith and O'Brien's (2006) network takes as input a lexeme identifier and a paradigm cell from the Spanish conjugation, to output the correct realisation for the corresponding form. However, the output is not a full form, but an identifier for one of a predefined set of possible realisations: e.g. *-amos* for CANTAR 'sing', PRES IND, 1P. A more psycholinguistically inspired connectionist network is described by Thymï¿œ et al. (1994), which nonetheless applies to a few noun lemmas only, and is mostly intended to simulate human errors. Finally, Temporal Self-Organising Maps (TSOMs) have recently been proposed as models of dynamic memories for symbolic time-series. In TSOMs, words are represented as chains of specialised processing nodes, that selectively fire when specific symbols are input in specific temporal contexts. Node specialisation is the outcome of the interplay of two training principles, based on entrenchment and competition. Marzi and colleagues (2018) discuss some important properties of TSOMs trained on 6 inflection systems of different complexity (including one typologically different one). However plausible as models of abstractive paradigm-based learning of inflection morphology, TSOMs are, however, not readily amenable to being used for the cell-filling problem.

Here, we consider another different connectionist architecture, based on Long Short Term Memories (LSTMs), recently proposed by Malouf (2016, 2017) to address the cell-filling problem.

3. The Experiment

Cell-filling can be simulated by training a learning model on a number of partial paradigms, to then complete them by generating missing forms. Training consists of <lemma_paradigm cell, inflected form> pairs. A lemma is not a form (e.g. *credere* 'to believe'), but a symbolic proxy of its lexical content (CREDERE). Word inflection consists of producing a fully inflected form given a known lemma and an empty paradigm cell. It is important to appreciate that for a model to be able to produce an inflected form on the basis of a cluster of lexical and morpho-syntactic features, it has to learn the symbol sequence making up the stem of the verb in question in the first place and to combine it with the paradigmatically appropriate inflectional ending.

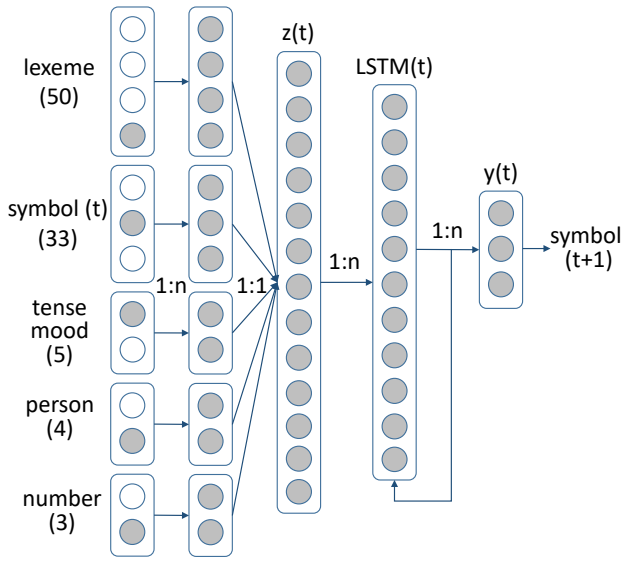

Figure 1
The network architecture. The input vector dimension is shown in brackets. Trainable dense projection matrices are shown as $1:n$, and concatenation as $1:1$.

3.1 Methods and materials

Following Malouf (2017), the LSTM network in Figure 1 is designed to take as input a lemma (e.g. CREDERE), a set of morpho-syntactic features (e.g. PRES_IND, 3, S) and a sequence of symbols (<*crede*>)[1] one symbol s_t at a time, to output a probability distribution over the upcoming symbol s_{t+1} in the sequence: $p(s_{t+1}|s_t$,CREDERE, PRES_IND, 3, S). To produce the form <*crede*>, we take the start symbol '<' as s_1, use s_1 to predict s_2, then use the predicted symbol to predict s_3 and so on, until '>' is predicted. Input symbols are encoded as mutually orthogonal one-hot vectors with as many dimensions as the overall number of different symbols used to encode all forms in the dataset.

The architecture was implemented in the Python language using two software libraries: Keras[2] and TensorFlow[3]. Keras is a high-level Python library that allows to define artificial neural networks using simple (even functional) APIs. Once defined the neural architecture, Keras relies on other libraries for all the numerical computations. In our case, we used the library TensorFlow (and its Python API) configured to run on the GPU. [4]

Unlike in Malouf's architecture, where morpho-syntactic features are holistically encoded in the input layer as one-hot vectors, each representing an orthogonal bundle of tense, mood, person and number features, here the morpho-syntactic features of tense, person and number are given independent one-hot vectors, whose dimensions equal the

1 '<' and '>' are respectively the start-of-word and the end-of-word symbols
2 `https://keras.io/`
3 `https://www.tensorflow.org/`
4 When running on a nVidia GeForce GTX970 with 1664 CUDA cores, a single training iteration (one folder in the leave-one-out cross-validation, with mini-batches of size 32) lasts between 15 and 30 seconds, depending on the network size and on the training language (more precisely, on the average length of the input sequences of symbols).

Table 1
Language datasets. Form length is measured by the number of orthographic symbols. In the Italian sample, the orthographic accent is encoded as a separate character (e.g. *ï¿œ* = e'). Differences between form types and cardinality of the training set are due to syncretism (particularly extensive in English).

language	*min/max form length*	*regular/irregular paradigms*	*form types/ training size*
English	2/11	20/30	208/750
German	3/11	16/34	504/750
Italian	2/12	23/27	748/750
Spanish	2/15	23/27	715/750

number of different values each feature can take. An extra dimension is added when a feature can be left uninstatiated in particular forms, as is the case with person and number features in the infinitive. No information is given about conjugation class for those languages (like Italian and Spanish) with more than one such class. This choice is motivated by the need to keep our network architecture as language-independent as possible, thus minimising recourse to those representational "tricks" that presuppose some knowledge of the language being learned. Language morphologies may differ considerably in the way morpho-syntactic feature bundles (e.g. <PRES_IND, 3, S>) are mapped onto surface markers. Some (more fusional) languages appear to realise a whole bundle of features with a single marker, whereas more agglutinative languages assign different markers to different features in the same bundle. Accordingly, a more distributed representation of morpho-syntactic features on the input layer is the most uncommitted, language-neutral option. Preliminary tests of the network performance using either style of feature encoding (i.e. "bundled" vs. "distributed") confirmed this assumption, showing that Malouf's encoding style is linguistically more biased than a distributed encoding of morpho-syntactic features is.

All input vectors are encoded by trainable dense matrices whose outputs are concatenated into the projection layer $z(t)$, which is input, in turn, to a layer of LSTM blocks (Figure 1). The LSTM layer takes as input both the information of $z(t)$, and its own output at *t–1*. Recurrent LSTM blocks are known to be able to capture long-distance relations in time series of symbols (Bengio, Simard, and Frasconi 1994; Hochreiter and Schmidhuber 1997; Jozefowicz, Zaremba, and Sutskever 2015), avoiding classical problems with training gradients of Simple Recurrent Networks (Jordan 1986; Elman 1990).

We tested our model on four comparable sets of English, German, Italian and Spanish inflected verb forms (Table 1), where paradigms are selected by sampling the highest-frequency fifty paradigms in large monolingual reference corpora (the Celex database for German and English (Baayen, Piepenbrock, and Gulikers 1995), the Italian Paisï¿œ corpus (Lyding et al. 2014), the European Spanish Subcorpus of the Spanish TenTen Corpus (www.sketchengine.co.uk)). For all languages, a fixed set of cells was chosen from each paradigm: all present indicative forms (1SIE, 2SIE, 3SIE, 1PIE, 2PIE, 3PIE), all past tense forms (1SIA, 2SIA, 3SIA, 1PIA, 2PIA, 3PIA), infinitive (i), past participle (pA), German and English present participle/Italian and Spanish gerund (pE). Each training form was administered once per epoch, and training was stopped when the training accuracy did not improve by more than a fixed threshold (results in this paper have

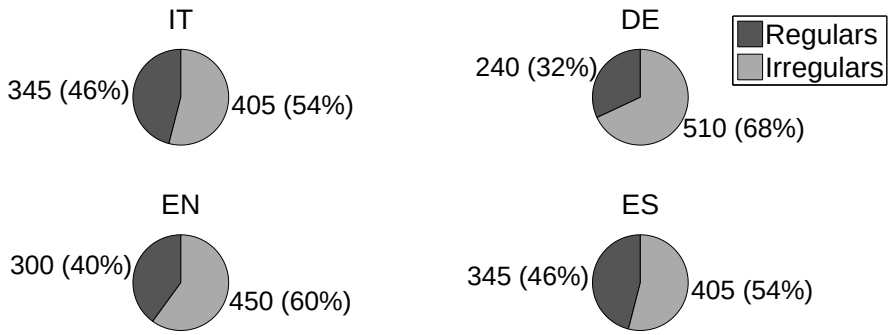

Figure 2
Composition of the four datasets (IT: Italian, DE: German, EN: English, ES: Spanish) in terms of percentage of forms belonging to regular and irregular paradigms. As expected, for all languages, the majority of top-frequency paradigms are irregular.

been obtained using the value 0.005) for a number consecutive "patience" epochs (set to five in these experiments). Although a uniform distribution is admittedly not realistic, it increases the entropy of the cell-filling problem, to define some sort of upper bound on the complexity of the task.

The four sets exhibit extensive stem allomorphy and a rich set of affixations, including circumfixation (German *ge-mach-t* 'made', past participle). Most importantly, the distribution of stem allomorphs is accountable in terms of equivalence classes of cells, forming morphologically heterogenous, phonologically poorly predictable, but fairly stable sub-paradigms (Pirrelli 2000). Selection of the contextually appropriate stem allomorph for a given cell thus requires knowledge of the form of the allomorph and of its distribution within the paradigm. For example, that 1S, 3S and 3P cells of the Italian PASSATO REMOTO always select the same stem (*e.g. pres-i*, *pres-e* and *pres-ero* of PRENDERE 'take') is a general property of the Italian conjugation system. Similarly, if an irregular English paradigm presents two stem allomorphs (say stem_1 = *find* and stem_2 = *found*), stem_2 is selected in all past tense and past participle cells, whereas stem_1 is selected elsewhere. Finally, of the four verb systems, German, Italian and Spanish present a wide variety of inflectional endings, with the German set of endings being smaller and more systematic than the other two. Among all test languages, the English verb system is of a more isolating type, with a considerably more restricted set of inflectional endings, and a plethora of bare stem forms, i.e. inflected forms with zero affixation.

4. Results

To provide a useful benchmark for the performance of the LSTM network on the cell-filling task, we used the baseline system for Task 1 of the CoNLL-SIGMORPHON-2017 Universal Morphological Reinflection shared task.[5] The model changes the base form of a verb lemma into its inflected forms through rewrite rules of increasing specificity,

5 `https://github.com/sigmorphon/conll2017` (written by Mans Hulden).

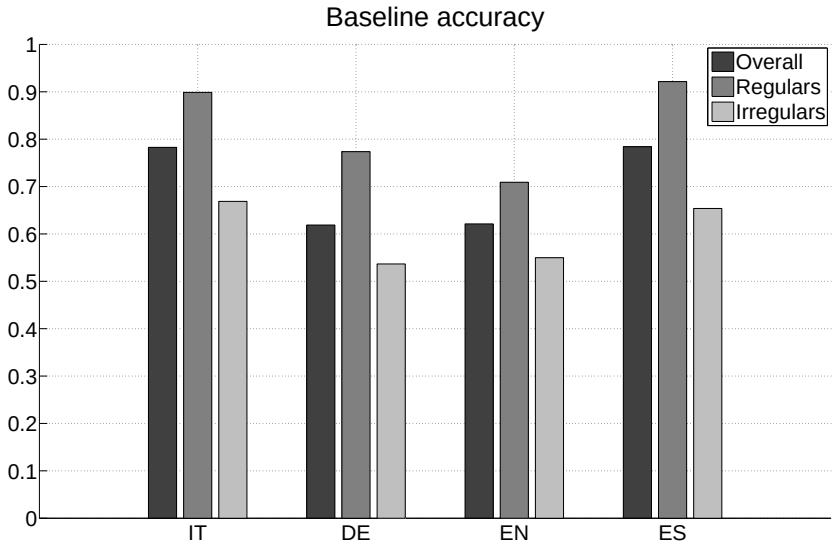

Figure 3
Per-word accuracy of the CoNLL baseline model tested on Italian (IT), German (DE), English (EN) and Spanish (ES).

automatically discovered from annotated training data. Base forms are infinitive forms for German, Italian and Spanish, and bare stems for English. To illustrate, two Italian forms such as *badare* 'to look after' and *bado* 'I look after' stand in a BASE :: PRES_IND_3S relation. The most general rule changing the former into the latter is *-are -> -o*, but more specific rewrite rules can be extracted from the same pair: *-dare -> -do*, *-adare -> -ado*, *-badare -> -bado*. The algorithm then generates the PRES_IND_3S of - say - *diradare* 'thin out', by using the most specific rewrite rule, i.e. the rewrite rule with the longest left-hand side matching *diradare* (namely *-adare > -ado*). If there is no matching rule, the base is used as a default output. It should be appreciated that the task is considerably simpler than the cell-filling problem. The algorithm can in fact infer, from the input, information about the verb base form (and, for Italian and Spanish, also about the verb conjugation class), which is to be learned from scratch in the context of the cell-filling problem.

In assessing the accuracy of the two learners, we used a leave-one-out cross-validation protocol: each form was left out of the training set, and predicted on the basis of all remaining forms. For LSTMs, this required running 750 networks, each trained on 749 exemplars and tested on the left-out exemplar.

The CoNLL baseline algorithm proves to be fairly effective for all test languages, but unevenly so (Figure 3). Somewhat unexpectedly, the model performs better on the more paradigmatically complex systems (namely Italian and Spanish, see final discussion), where regulars are inferred with remarkable accuracy (89.86% for Italian, and 91.31% for Spanish). Irregulars are predicted consistently worse (62.47% and 60.49% respectively). Accuracy on regulars is 77.92% for German and 73.67% for English, with a drop for irregulars (accuracy 51.37% and 54.89% respectively). By comparison, LSTM networks of different block size perform, on average, better than the baseline model (Figure 4). Once more, comparative accuracy varies with languages. Note that the most accurately predicted language by the baseline model (Italian) is the least accurately predicted by

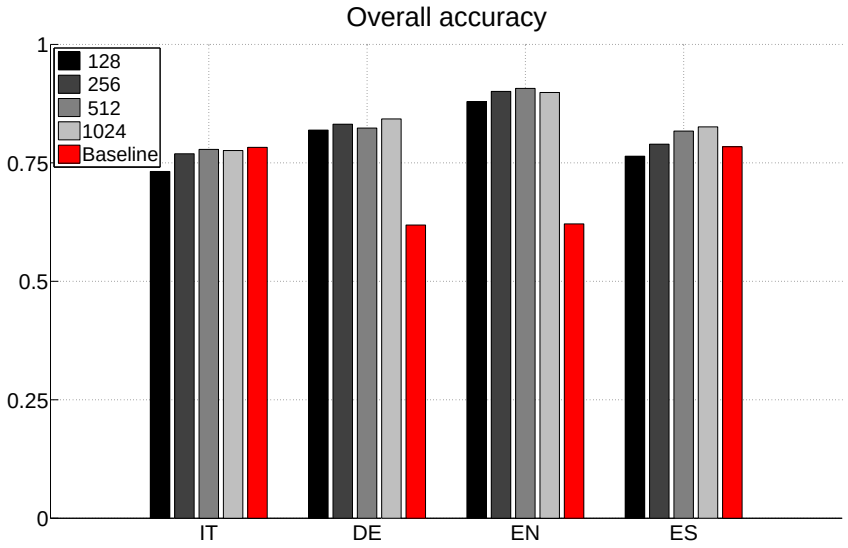

Figure 4
Per-word accuracy for Italian (IT), German (DE), English (EN) and Spanish (ES). Overall test scores are given for all LSTM network types (ordered by increasing number of LSTM blocks) and the CoNLL baseline.

LSTM networks. Nevertheless, even in this LSTM worst case, LSTM accuracy is at about the same level of the baseline accuracy. This is also true for Spanish. As for German and English, LSTM networks fare consistently better than the baseline model does (Figure 4). LSTMs average accuracy is: 73.30% on Italian, 76.77% on Spanish, 82.50% on German, and 90.20% on English forms. The CoNLL baseline accuracy is: 75.06% on Italian, 74.67% on Spanish, 59.87% on German, and 62.40% on English forms.

To check stability of our results, which could be heavily affected by LSTM initialisation point (Reimers and Gurevych 2017), we ran four different instances of the same experiment for each language, using the best performing block configuration given the language (namely 1024 for Italian, and 512 for all other languages). Accuracy scores averaged over the four istances are as follows: 74.30% on Italian (*sd* 0.4%), 78.23% on Spanish (*sd* 1%), 82.6% on German (*sd* 0.9%), and 90.93% on English (*sd* 1.3%), confirming the stability of our results.

Finally, even in case of errors at the word level, LSTMs proves to approximate the missed target more closely than the baseline does, as shown by the average edit distance between targets and outputs for all systems (Figure 5).

5. Discussion

The Cell-filling problem is a non trivial language learning task, especially in a situation where the speaker is exposed to a sample of the most frequent verb paradigms only, which include the most irregular ones in the conjugation system of all our test languages (Table 1). In a few cases, the problem has admittedly no solution. For example, in our test languages it is simply impossible to predict the first person singular of the present indicative of the auxiliary BE, on the basis of information from all remaining cells of the

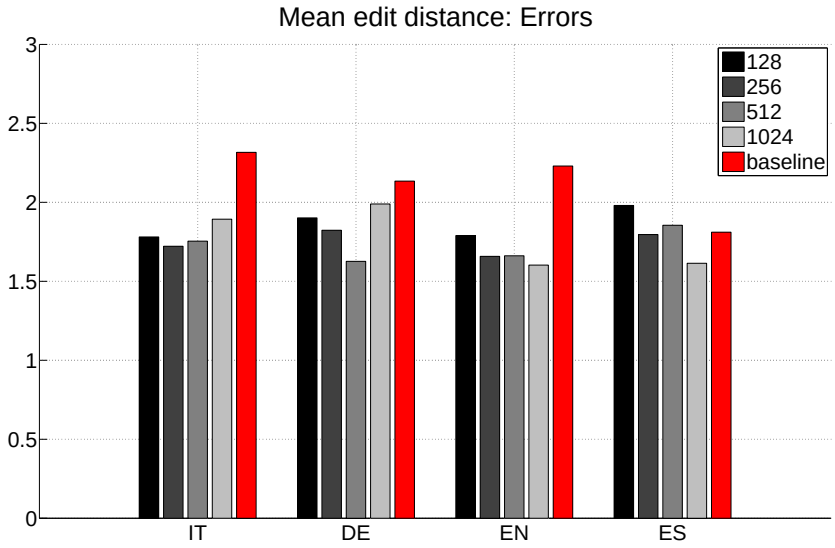

Figure 5
Mean edit distance between target form and output form for Italian (IT), German (DE), English (EN) and Spanish (ES). Distances are given for all LSTM network types (ordered by increasing number of LSTM blocks) and the CoNLL baseline.

paradigm. Another case in point is the German past participle *gefunden* 'found', which is the only form of FINDEN containing the stem alternant *fund-*, and has no other form with the same alternation pattern in our dataset. In fact, here we were not interested in replicating realistic and ecologically plausible learning conditions in child language maturation. Rather, we wanted to focus on the comparative difficulty of the task across a few languages, with the aim to assessing the ways in which LSTM networks address the logical problem of inferring novel inflected forms on the basis of the cumulative knowledge of their paradigm companions. Our experimental results clearly confirm the difficulty of the task. A powerful deep learning algorithm like an LSTM network compares well with the CoNLL baseline model, but it does not invariably outperform it. In this section, we discuss strengths and weaknesses of the two learning models.

The CoNLL baseline model is strongly reminiscent of Albright and Hayes' (2003) Minimal Generalisation Learner (MGL). MGL was, in fact, originally designed to infer Italian infinitives from first singular present indicative forms (Albright 2002). In the CoNLL model, inference goes in the opposite direction: from base forms to all other paradigm cells. Incidentally, this inferential relation is much more informative than the one adopted by Albright,[6] as the Italian infinitive contains information about the verb's thematic vowel, which is neutralised in the first singular present indicative forms, where the thematic vowel cancels out.

The CoNLL model has a clear analogy-based bias. Verb stems ending in the same way (compare - say - Italian *tend-ere* 'tend', *rend-ere* 'turn', and *prend-ere* 'take') undergo

6 In (Albright 2002), choice of the first person singular in the Italian present indicative was motivated precisely by the goal to investigate how easily the appropriate conjugation class of an Italian verb form can be predicted in those contexts where information of the thematic vowel is missing.

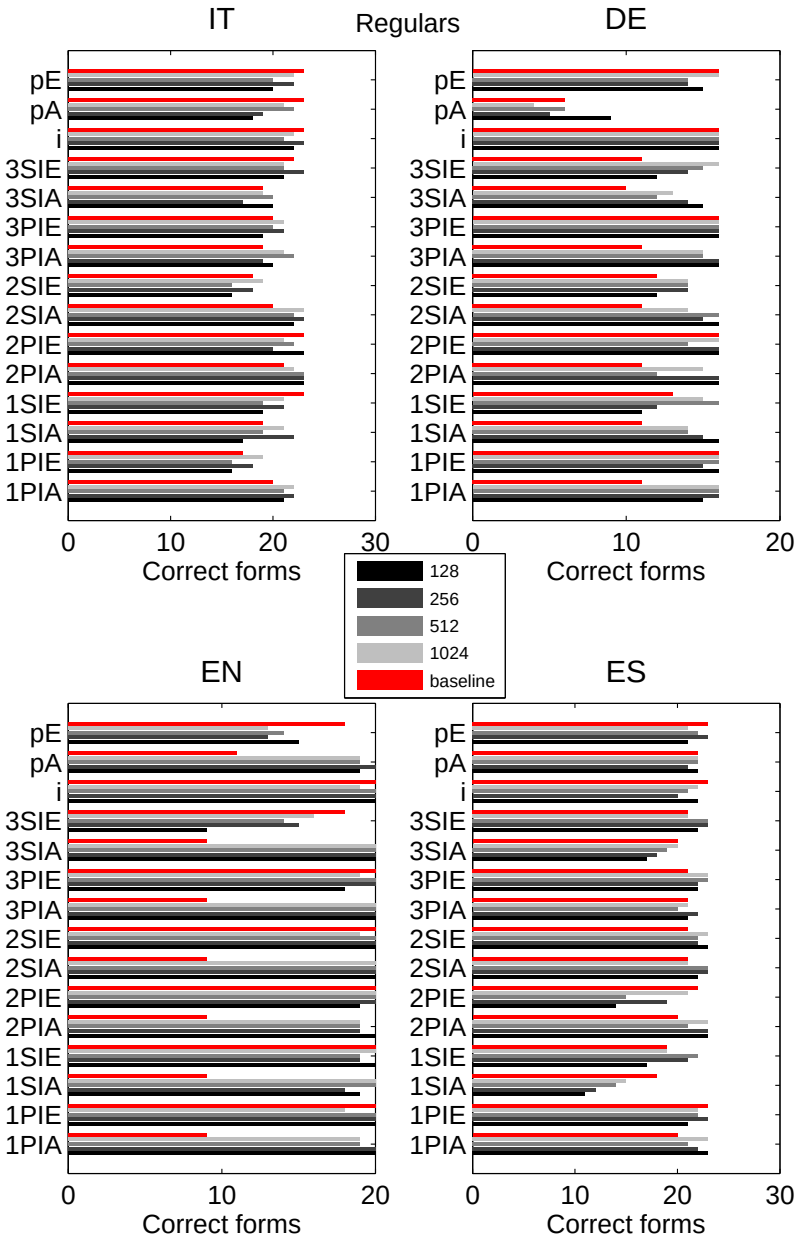

Figure 6
Per-cell accuracy of LSTMs and CoNLL baseline on regular paradigms in the four test languages.

the same allomorphic readjustment (respectively, *teso* 'tended', *reso* 'turned' and *preso* 'taken' in the past participle). In Italian, unpredictable stem allomorphy affects, in the vast majority of cases, the final part of the verb stem, and it is historically motivated by the operation of local phonological rules (Burzio 2004; Pirrelli 2000). Overall, the number of stems undergoing the same irregular stem formation processes is com-

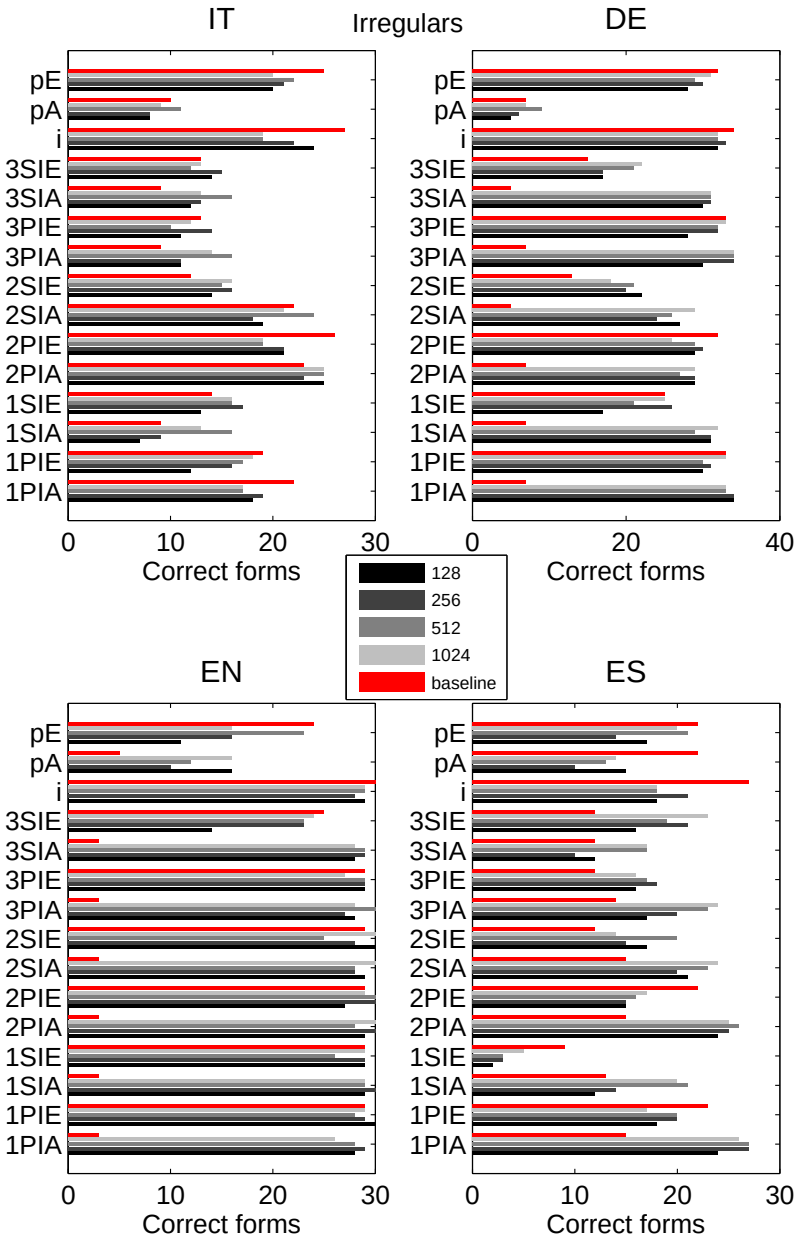

Figure 7
Per-cell accuracy of LSTMs and CoNLL baseline on irregular paradigms in the four test languages.

paratively large. Such a densely populated analogical space makes it highly likely that an unknown target form (e.g. *preso*) undergoes the same allomorphic process of the paradigmatically corresponding form of another paradigm (e.g. *reso*) whose base stem (*rend-*) ends in the same way as the target base stem (*tend-*). This makes the CoNLL baseline a powerful generalisation algorithm. Unsurprisingly, the same local

generalisation strategy is successful for Spanish too. However, in both languages, it fails to deal with stem allomorphy involving vowel apophony (as in *esco/usciamo* 'I go out/we go out'), or diphthongisation (as in Italian *vengo/vieni* 'I come/you (2S) come', and Spanish *cuento/contar* 'I count/to count', *vuelvo/volver* 'I come back/to come back'), where phonological changes are not limited to the characters immediately preceding the stem boundary.

On the other hand, it turns out that the CoNLL baseline is a much weaker learner of German and English, where stem allomorphy is not confined to the stem boundary, and is more consistently distributed throughout the paradigm. This is shown in Figures 6 (for regular paradigms) and 7 (for irregular paradigms), plotting the per cell accuracy of LSTMs and the baseline for all test languages. In all past tense and past participle cells of both German and English, where stem allomorphy applies systematically, the baseline algorithm is considerably less accurate than LSTMs are. To illustrate, let us focus on how the baseline algorithm infers the past participle of SAY. Our training set contains one maximally overlapping stem: PLAY. However, this is a "false" analogical friend to SAY, yielding the wrong past participle form **sayed*. This shows a general over-regularisation bias of the baseline, extensively witnessed by most (irregular) ablauting forms in German, which are over-regularised by the CoNLL baseline (e.g. **nemmst* for *nimmst* 'you take', *gebt* for *gibt* 'it gives').

For sure, over-regularised outcomes are plausible and frequent in child production of inflection (Clahsen, Hadler, and Weyerts 2004). A larger set of training exemplars than the one used here, including, e.g., forms of PAY (past participle *paid*), is expected to provide the evidence needed to inflect SAY correctly. Be that as it may, for our present purposes, a detailed comparison of the results of LSTMs with the CoNLL baseline is useful to understand more of the underlying morphological structure of our training data, as well as LSTMs' learning bias.

5.1 Regulars vs. irregulars

In dealing with German and English irregularly inflected forms, the purely syntagmatic approach of the CoNLL baseline, deriving all inflected forms from an underlying base, is too surface-oriented and misses some significant non local constraints. Simply put, the orthotactic/phonotactic structure of Germanic stems is less criterial for stem allomorphy than the orthotactic/phonotactic structure of Romance stems. A much more robust generalisation strategy for the two Germanic languages is to exploit the larger formal paradigmatic syncretism of their verb system.

Although LSTMs have no information about the morphological structure of input forms, they are considerably more robust than our baseline in this respect. Memory resources allowing, LSTMs appear to keep track of two types of syntagmatic constraints: short range phonological/orthotactic patterns (preventing the network to output phonotactically implausible strings such as **seemng* for target *seeming*), as well as longer range morphotactic constraints, covering the sequential structure of prefixes, stems and suffixes. Another, related issue of some theoretical interest, is to assess whether LSTMs are able to enforce global, paradigmatic constraints, whereby *all* paradigmatically-related forms contribute to fill in gaps in the same paradigm. In the end, knowledge that a paradigm contains a few stem allomorphs is good reason for a speaker to produce a stem allomorph in other (empty) cells. The more systematic the distribution of stem alternants is across the paradigm, the easier for the speaker to fill in empty cells. In this respect, German and English conjugations prove to be

paradigmatically well-behaved. Several pieces of evidence show that LSTMs are able to discover at least a few syntagmatic and paradigmatic redundancies of this kind.

First, output paradigm cells play a distinctive role in driving the generalisation bias of LSTMs. In the baseline model, a target inflected form is produced by finding the base analogue in the training data that best fits the target base. Whenever a target inflected form is produced on the basis of the best analogue to its base, the same form is output in all other cells where the best analogue happens to have identical forms. For example, if BASE::SAY best matches BASE::PLAY, SAY will be inflected as **sayed* in all part participle and past tense cells, where PLAY is inflected as *played*. This is because MGL generalisation is based on matching *input conditions* only. This is not the case for LSTMs. For example, the form **maken* is wrongly produced for PAST_PART::make, but *made* is correctly output for all past tense forms. This provides a strong indication that generalisation is also based on output cells, not on input conditions only.

A second related issue is whether LSTMs can develop global constraints on the distribution of stem allomorphs across the paradigm. We find some evidence that this is the case in analysing patterns of errors. Occasionally, past tense forms are wrongly output in past participle cells. So we find *wrote* for *written*, *took* for *taken* and *began* for *begun*. However, this pattern of errors is rather unsystematic. We suspect that it may simply be due to the repeated association of past tense forms with the feature PAST in the input vector.

What role does morphological structure play in the LSTM generalisation bias? Due to the predictive nature of the production task and the LSTM re-entrant layer of temporal connectivity, the network develops a left-to-right sensitivity to upcoming symbols, with per-symbol accuracy being a function of the network confidence about the next output symbol.[7] As we saw, this sensitivity to sequential patterns is responsible for the network's control on orthotactically/phonotactically plausible sequences. To assess the correlation between per-symbol accuracy and "perception" of morphological structure, we used Generalised Additive Models (GAM) interpolating the "average" accuracy of the learning algorithm in producing an upcoming symbol as a function of the symbol position to the inflectional boundary of each form. Results are not unequivocal, as illustrated in Figure 8.

The regression plots of Figure 8 show a clear structural effect of the distance to the stem-ending boundary of LSTM output symbols in German, Italian and Spanish, where accuracy drops to its minimum value around the morpheme boundary (corresponding to the 0 value on the x axis of the regression plots). In Italian, the most apparent difference between LSTMs (dotted lines) and the baseline (red solid line) is observed across the inflectional endings, where the per-symbol accuracy of the baseline is significantly lower than the accuracy of 256, 512 and 1028 block LSTMs. Nonetheless, per-word accuracy scores on Italian are higher in the CoNNL baseline than in LSTMs. In Spanish, the 256, 512, 1028 block LSTMs perform consistently better than the baseline. In German, the overall advantage of LSTMs over the baseline is marked by the characteristically U-shaped curve of per-symbol accuracy at the stem-ending boundary. Once more, this position marks a point of structural discontinuity in inflected verb forms. Intuitively, production of an inflected form by an LSTM network is fairly easy at the beginning of

7 For each position in the target verb form, a matching output letter is given a score of 1, and a non matching output letter a score of 0. An average score of 1 in the model means that the letter in that position is always correctly predicted, and an average score of 0 means that it is always missed.

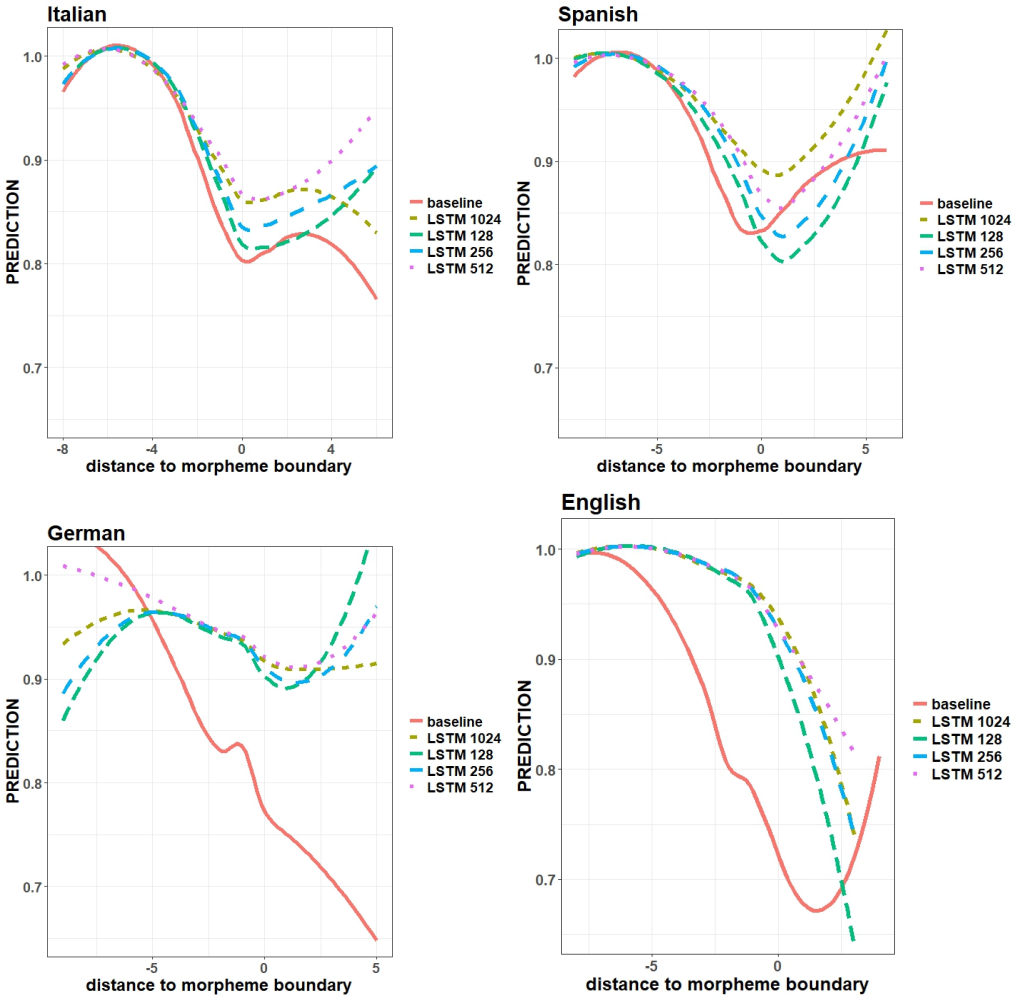

Figure 8
Regression plots of the interaction between distance to morpheme boundary (between stem and inflectional ending) and learning model in a GAM fitting per-symbol prediction accuracy in the four test languages.

the stem, but it soon gets more difficult when approaching the morpheme boundary, particularly with irregulars.

On the other hand, the English plot provides little or no evidence of structure sensitivity. The apparent U-shaped profile of the baseline does not denote a greater accuracy on long inflection endings relative to LSTM accuracy. It is merely a nonlinear interpolation effect making up for the poor performance of the baseline algorithm on English (irregular) stems compared with the corresponding endings. In fact, all LSTM models are significantly more accurate than the baseline on a long inflectional ending such as *-ing* (Figure 9, left).

The evidence has non trivial implications from a typological point of view. In a comprehensive comparison of nearly two dozen languages (in the Indo-European, Ugro-Finnic and Semitic families plus Turkish), Bittner and colleagues (2003) arrive

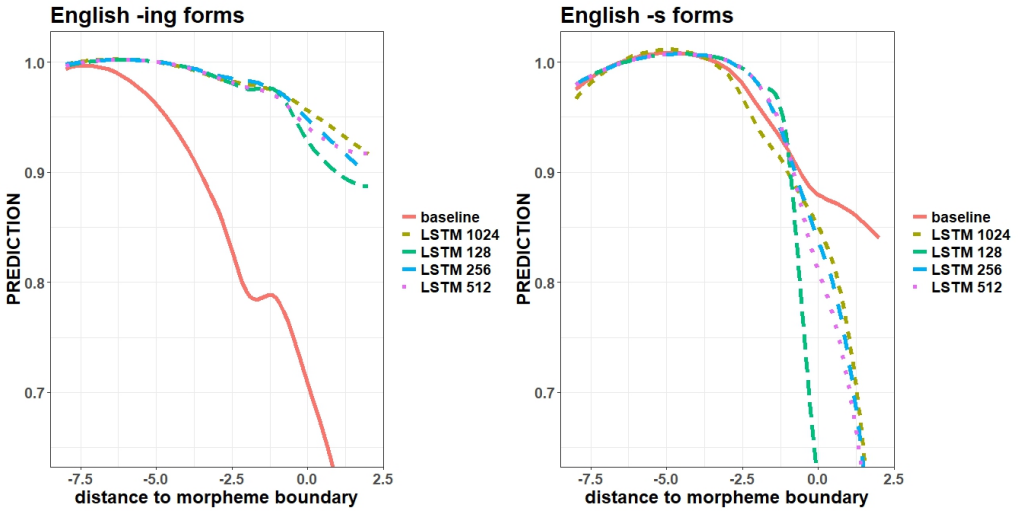

Figure 9
Regression plots of the interaction between distance to morpheme boundary (between stem and inflectional ending) and learning model in a GAM fitting per-symbol prediction accuracy in English *ing*-forms (left) and *s*-forms (right).

at the conclusion that acquisition of inflection is crucially conditioned by typological factors such as richness, uniformity and transparency of inflectional paradigms. They provide the following schema, where some European languages are arranged along a typological continuum ranging from the inflecting-fusional type (left) to the more isolating type (right):

Lithuanian→Greek→Russian→Croatian→Italian→
Spanish→German→Dutch→French→English.

The implication of this schema for our concerns is that an English inflected form provides, as such, little evidence of structural discontinuity. It is then to be expected that we find sparser evidence of processing uncertainty for symbol prediction at the morpheme boundary in a more isolating language like English. This is confirmed by evidence from child language acquisition of English inflection (Haegeman 1995; Phillips 1996), showing that English children tend to omit *s*-marking in the realisation of present indicative third singular forms, due to the overwhelming pressure of base forms in the same subparadigm. LSTMs, unlike CoNLL baseline, show a similar behaviour (Figure 9, right).[8] Note, finally, that the typological hierarchy above is somewhat mirrored by the accuracy results we obtained with LSTMs (Figure 4), where Italian is the most difficult language to be generalised over in production, and English is the easiest one. The CoNLL baseline does not seem to follow the same hierarchy. That more irregular paradigms are more difficult to learn appears to match the intuition that the morphologies of some languages are more complex than those of other languages.

8 This typological effect is somewhat amplified by the criteria we adopted for defining the position of the stem-ending boundary. For example, following Aronoff (1994), we consider an irregular past participle like English *made* as a full allomorphic stem, with no ensuing affixation.

6. Concluding remarks

The cell-filling problem addresses the ecological, developmentally motivated task of inferring novel inflected forms based on evidence of familiar forms. Other (simpler) models have been proposed in the literature to account for form-meaning mapping in Morphology (Baayen et al. 2011; Plaut and Gonnerman 2000, among others). Nevertheless, we do not know of any other artificial neural networks that can simulate word inflection as a cell-filling task. Unlike more traditional connectionist architectures (Rumelhart and McClelland 1987), recurrent LSTMs do not presuppose the existence of underlying base forms, but they learn possibly alternating stems upon exposure to linguistically annotated full forms. Admittedly, the use of orthogonal one-hot vectors for lemmas, unigram temporal series for inflected forms, and abstract morpho-syntactic features as a proxy of context-sensitive functional agreement effects, are crude representational short-hands. Nonetheless, in tackling the task, LSTMs prove to be able to orchestrate different sources of word knowledge, well beyond pure surface word relations: namely morphological structure (stem-affix boundaries), paradigm organisation and degrees of (ir-)regularity in stem formation. Acquisition of different inflectional systems may require a different balance of all these pieces of knowledge.

Unlike more *ad hoc* algorithms, LSTMs appear to be flexible and powerful enough to be able to adapt their learning strategy to the specific properties of inflectional systems of different complexity. This strikes us as an important bonus of LSTMs. In addressing a task like the cell-filling problem, which does not seem to require information about very long sequences of input symbols, LSTMs prove to be able to discover other complex constraints than just sequential or syntagmatic ones, using memory of input forms in complementary distribution as global generalisation patterns. Having said that, we should also emphasise that that the cell-filling problem turned out to be a rather recalcitrant and challenging task even for a powerful machine learning technology like LSTMs. We gathered sparse evidence that LSTMs can develop a stem variable capturing the sweepingly systematic patterns of stem distribution across paradigm cells. Like more traditional associative connectionist networks, it looks like LSTMs can learn a universally quantified one-to-one mapping relation only if this relation is illustrated with respect to each possible input/output pairs (Marcus 2001). Hence, even when an LSTM network is exposed to a number of paradigms instantiating the same pattern of stem distribution, the pattern is not readily extended to the unknown form of a partially filled in paradigm. We expect more experiments on typologically more diverse languages to be needed before the issue of the cognitive plausibility of LSTMs as models of the human word processor can be assessed on a firmer empirical basis.

References

Ackerman, Farrell, James P. Blevins, and Robert Malouf. 2009. Parts and wholes: Patterns of relatedness in complex morphological systems and why they matter. In James P. Blevins and Juliette Blevins, editors, *Analogy in Grammar: Form and Acquisition*. Oxford University Press.

Ackerman, Farrell and Robert Malouf. 2013. Morphological organization: The low conditional entropy conjecture. *Language*, 89(3):429–464.

Albright, Adam. 2002. Islands of reliability for regular morphology: Evidence from italian. *Language*, 78(4):684–709.

Albright, Adam and Bruce Hayes. 2003. Rules vs. analogy in english past tenses: A computational/experimental study. *Cognition*, 90(2):119–161.

Aronoff, Mark. 1994. *Morphology by itself: Stems and inflectional classes*. Number 22.

Baayen, R. Harald, Petar Milin, Dusica Filipović Đurđević, Peter Hendrix, and Marco Marelli. 2011. An amorphous model for morphological processing in visual comprehension based on

naive discriminative learning. *Psychological review*, 118(3):438–481.
Baayen, R. Harald, Richard Piepenbrock, and Leon Gulikers, 1995. *The CELEX Lexical Database (CD-ROM)*. Linguistic Data Consortium, University of Pennsylvania, Philadelphia, PA.
Bengio, Yoshua, Patrice Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166.
Bickel, Balthasar and Johanna Nichols. 2005. Inflectional synthesis of the verb. In David Gil Martin Haspelmath, Matthew S. Dryer and Bernard Comrie, editors, *The World Atlas of Language Structures*. Oxford University Press, pages 94–97.
Bittner, Dagmar, Wolfgang U. Dressler, and Marianne Kilani-Schoch, editors. 2003. *Development of Verb Inflection in First Language Acquisition: a cross-linguistic perspective*. Mouton de Gruyter, Berlin.
Blevins, James P. 2006. Word-based morphology. *Journal of Linguistics*, 42(3):531–573.
Blevins, James P., Petar Milin, and Michael Ramscar. 2017. The zipfian paradigm cell filling problem. In Ferenc Kiefer, James P. Blevins, and Huba Bartos, editors, *Morphological Paradigms and Functions*. Brill, Leiden.
Burzio, Luigi, 2004. *Paradigmatic and syntagmatic relations in Italian verbal inflection*, volume 258, pages 17–44. John Benjamins, Amsterdam-Philadelphia.
Clahsen, Harald, Meike Hadler, and Helga Weyerts. 2004. Speeded production of inflected words in children and adults. *Journal of child language*, 31(3):683–712.
Colombo, Lucia, Alessandro Laudanna, Maria De Martino, and Cristina Brivio. 2004. Regularity and/or consistency in the production of the past participle? *Brain and language*, 90(1):128–142.
Coltheart, Max, Kathleen Rastle, Conrad Perry, Robyn Langdon, and Johannes Ziegler. 2001. Drc: a dual route cascaded model of visual word recognition and reading aloud. *Psychological review*, 108(1):204.
Dąbrowska, Ewa. 2004. Rules or schemas? evidence from polish. *Language and cognitive processes*, 19(2):225–271.
Elman, Jeffrey L. 1990. Finding structure in time. *Cognitive Science*, 14(2):179–211.
Elman, Jeffrey L. 2009. On the meaning of words and dinosaur bones: Lexical knowledge without a lexicon. *Cognitive science*, 33(4):547–582.
Evans, Nicholas and Stephen C. Levinson. 2009. The myth of language universals: Language diversity and its importance for cognitive science. *Behavioral and Brain Sciences*, (32):429–92.
Goldberg, Adele E. 2003. Constructions: a new theoretical approach to language. *Trends in cognitive sciences*, 7(5):219–224.
Goldsmith, John. 2001. Unsupervised learning of the morphology of a natural language. *Computational linguistics*, 27(2):153–198.
Goldsmith, John and Jeremy O'Brien. 2006. Learning inflectional classes. *Language Learning and Development*, 2(4):219–250.
Haegeman, Liliane. 1995. Root infinitives, tense, and truncated structures in dutch. *Language acquisition*, 4(3):205–255.
Harm, Michael W. and Mark S. Seidenberg. 1999. Phonology, reading acquisition, and dyslexia: insights from connectionist models. *Psychological review*, 106(3):491.
Hochreiter, Sepp and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.
Jordan, Michael. 1986. Serial order: A parallel distributed processing approach. Technical Report 8604, University of California.
Jozefowicz, Rafal, Wojciech Zaremba, and Ilya Sutskever. 2015. An empirical exploration of recurrent network architectures. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 2342–2350, Lille, France, 07–09 July.
Keuleers, Emmanuel and Walter Daelemans. 2007. Memory-based learning models of inflectional morphology: A methodological case-study. *Lingue e linguaggio*, 6(2):151–174.
Lyding, Verena, Egon Stemle, Claudia Borghetti, Marco Brunello, Sara Castagnoli, Felice Dell'Orletta, Henrik Dittmann, Alessandro Lenci, and Vito Pirrelli. 2014. The paisá corpus of italian web texts. Proceedings of the 9th Web as Corpus Workshop (WaC-9)@ EACL 2014, pages 36–43, Gothenburg, Sweden, April, 26. Association for Computational Linguistics.
Malouf, Robert. 2016. Generating morphological paradigms with a recurrent neural network. *San Diego Linguistics Papers*, (6):122–129.
Malouf, Robert. 2017. Abstractive morphological learning with a recurrent neural network. *Morphology*, 27(4):431–458.

Marcus, Gary. 2001. *The algebraic mind*. MIT Press.
Marzi, Claudia, Marcello Ferro, Franco Alberto Cardillo, and Vito Pirrelli. 2016. Effects of frequency and regularity in an integrative model of word storage and processing. *Italian Journal of Linguistics*, 28(1):79–114.
Marzi, Claudia, Marcello Ferro, Oaufae Nahli, Patrizia Belik, Stavros Bompolas, and Vito Pirrelli. 2018. Evaluating inflectional complexity crosslinguistically: a processing perspective. In *Proceedings of the 11th edition of the Language Resources and Evaluation Conference*, Miyazaki (Japan), 7-12 May.
McClelland, James L. and David E. Rumelhart. 1981. An interactive activation model of context effects in letter perception: I. an account of basic findings. *Psychological review*, 88(5):375.
McNamee, Paul, Charles Nicholas, and James Mayfield. 2009. Addressing morphological variation in alphabetic languages. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 75–82, Boston, MA, USA, July 19 - 23. ACM.
McWorther, John. 2001. The world's simplest grammars are creole grammars. *Linguistic Typology*, (5):125–166.
Orsolini, Margherita, Rachele Fanari, and Hugo Bowles. 1998. Acquiring regular and irregular inflection in a language with verb classes. *Language and cognitive processes*, 13(4):425–464.
Orsolini, Margherita and William Marslen-Wilson. 1997. Universals in morphological representation: Evidence from italian. *Language and Cognitive Processes*, 12(1):1–47.
Pauw, Guy De and Peter Waiganjo Wagacha. 2007. Bootstrapping morphological analysis of gĩkũyũ using unsupervised maximum entropy learning. In *Eighth Annual Conference of the International Speech Communication Association*.
Perry, Conrad, Johannes C. Ziegler, and Marco Zorzi. 2007. Nested incremental modeling in the development of computational theories: the cdp+ model of reading aloud. *Psychological review*, 114(2):273.
Phillips, Colin. 1996. Root infinitives are finite. In *Proceedings of the 20th annual Boston University conference on language development*, pages 588–599.
Pirrelli, Vito. 2000. *Paradigmi in morfologia. Un approccio interdisciplinare alla flessione verbale dell'italiano*. Istituti Editoriali e Poligrafici Internazionali, Pisa.
Pirrelli, Vito and François Yvon. 1999. The hidden dimension: a paradigmatic view of data-driven nlp. *Journal of Experimental & Theoretical Artificial Intelligence*, 11(3):391–408.
Plaut, David C. and Laura M. Gonnerman. 2000. Are non-semantic morphological effects incompatible with a distributed connectionist approach to lexical processing? *Language and Cognitive Processes*, 15(4/5):445–485.
Plaut, David C., James L. McClelland, Mark S. Seidenberg, and Karalyn Patterson. 1996. Understanding normal and impaired word reading: computational principles in quasi-regular domains. *Psychological review*, 103(1):56.
Plunkett, Kim and Patrick Juola. 1999. A connectionist model of english past tense and plural morphology. *Cognitive Science*, 23(4):463–490.
Reimers, Nils and Iryna Gurevych. 2017. Reporting score distributions makes a difference: Performance study of lstm-networks for sequence tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 338–348, Copenhagen, Denmark. Association for Computational Linguistics.
Rumelhart, David E. and James L. McClelland. 1987. On learning the past tenses of english verbs. In David E. Rumelhart, James L. McClelland, and the PDP Research Group, editors, *Parallel Distributed Processing. Explorations in the Microstructures of Cognition*, volume 2 Psychological and Biological Models. MIT Press, pages 216–271.
Shosted, Ryan. 2006. Correlating complexity: a typological approach. *Linguistic Typology*, (10):1–40.
Thymé, Ann, Farrell Ackerman, and Jeff Elman, 1994. *Finnish Nominal Inflection. Paradigmatic Patterns and Token Analogy*, volume 26, page 445. John Benjamins Publishing Company.
Tomasello, Michael. 2000. The item-based nature of children's early syntactic development. *Trends in cognitive sciences*, 4(4):156–163.

CLiC-it 2017: A Retrospective

Roberto Basili*
Università di Roma, Tor Vergata

Malvina Nissim**
Rijksuniversiteit Groningen

Giorgio Satta†
Università di Padova

The Fourth Italian Conference on Computational Linguistics (CLiC-it 2017) took place in Rome, in December 2017. As in previous editions, it served as the prime forum in Italy for discussing research in computational linguistics and Natural Language Processing. As General Chairs, we offer a retrospective over the conference, highlighting its international flavour and its attention to students and young researchers, with a particular focus on the innovations that were introduced at the 2017 edition.

1. Context

The Fourth Italian Conference on Computational Linguistics (CLiC-it) took place on December 11–13, 2017, with more than 110 registered participants and for the first time in the wonderful city of Rome. The conference was locally organised by the University of Rome "Tor Vergata", and was hosted at the headquarters of the National Research Council of Italy (CNR). The CLiC-it conference series is an initiative of the Italian Association for Computational Linguistics (AILC) and, after four years of activity, it has clearly established itself as the premier national forum for research and development in the fields of Computational Linguistics and Natural Language Processing (CL/NLP), where leading researchers and practitioners from Italian academia and industry meet to share their research results, experiences, and challenges.

These annual meetings have explicitly allowed reports on ongoing research, with the goal of ensuring a wide participation of the community and motivated by an inclusive spirit. Usually, the number of submitted papers is over 60, with over 100 registered participants. A number of submissions are also accepted as poster presentations. To make participation even more attractive, some internationally well-known researchers from abroad are invited for a keynote lecture, and some panel events are also added to the program. Overall, this event structure has made it possible that "all" researchers in computational linguists in Italy meet together once every year.

2. Technical Program

The conference received 72 submissions, against 64 submissions in 2015 and 69 submissions in 2016. The Programme Committee worked very hard to ensure that every paper

* Department of Enterprise Engineering, University of Rome, Tor Vergata, Via del Politecnico, 1, 00133 Roma (Italy). E-mail: basili@info.uniroma2.it
** CLCG, Rijksuniversiteit Groningen, Groningen NL m.nissim@rug.nl
† Department of Information Engineering, University of Padua, Italy. E-mail: satta@dei.unipd.it

Table 1
Areas at the CLiC-it 2017, and number (proportion) of accepted papers per area.

AREA	ACCEPTED	
Cognitive Modeling of Language Processing and Psycholinguistics	3	5.2%
Information Extraction, Information Retrieval and Question Answering	3	5.2%
Language Resources	9	15.5%
Linguistic Issues in Computational Linguistics & Natural Language Processing	7	12.1%
Machine Learning and Language Processing	6	10.3%
Machine Translation and Multilinguality	3	5.2%
Morphology and Syntax Processing	2	3.4%
Natural Language Processing for Digital Humanities	6	10.3%
Natural Language Processing for Web and Social Media	6	10.3%
Pragmatics and Creativity	8	13.8%
Semantics and Knowledge Acquisition	3	5.2%
Spoken Language Processing and Automatic Speech Understanding	2	3.4%

received at least two careful and fair reviews. This process finally led to the acceptance of 21 papers for oral presentation and 37 papers for poster presentation, with a global acceptance rate of 80%, again in line with previous editions (81% in 2015 and 80% in 2016). Regardless of the format of presentation, all accepted papers are allocated 6 pages in the proceedings, available as open access publication.[1]

The conference was organised around 12 thematic areas that are basically the same as those of the 2016 edition of CLiC-it, with the only exception of the area Information Retrieval and Question Answering and the area Information Extraction, Entity Linking and (Linked) Open Data, which got merged. On the other hand, at this edition the conference implemented a considerable reduction on the number of area chairs, moving from 30 area chairs in 2016, with two or three area chairs per area, to 16 area chairs in 2017, with one or two area chairs per area, on the basis of the expected number of submissions. On a retrospect, the upper bound of two area chairs per area proved to be a reasonable one, given that the most populated area (Language Resources) received 13 submissions.

In Table 1 we show an overview of all of the thematic areas at CLiC-it 2017, along with the number of accepted papers and their proportion to the whole.

The most successful thematic area, in terms of number of accepted papers, has been Language Resources. Several novel datasets were presented, involving also languages other than Italian, such as English, German and Latin. Two other successful areas are Pragmatics and Creativity, which has covered work on social media, gender analysis, hate speech, irony detection and modality, and the area of Linguistic Issues, which has covered work on lexical semantics, idioms, phrase structure and syntactic selection. Among the remaining areas, very interesting work on deep learning for natural language processing has been presented in Machine Learning.

The Web and Social Media area was characterised mostly by works on affective computing, reporting on specific phenomena, resource creation, and predictive modelling, such as predictions about the Sanremo music festival competition, or controver-

1 Accademia University Press: `http://www.aaccademia.it/scheda-libro?aaref=1186`;
CEUR Workshop Proceedings, AI*IA Series: `http://ceur-ws.org/Vol-2006/`;
OpenEdition Books: `https://books.openedition.org/aaccademia/2314?lang=it`.

sies on Facebook. Finally several works in Digital Humanities focused on a wide variety of data, including a stylometric analysis of the Talmud, and the diachronic distribution of certain noun classes in Latin.

Overall, the conference provided an intellectually stimulating environment for the exchange of ideas, with a broad range of subjects being investigated, as well as a very lively picture of the Italian community working in the field, with nearly half of the participants being PhD students or else junior researchers. The conference also managed to attract a few papers from private companies, including large industrial groups, attesting growing interest for the field also outside of the Italian academia.

3. Worldwide Computational Linguistics at CLiC-it 2017

While CLiC-it is the conference of the Italian Association for Computational Linguistics, it also aims at achieving an international stand. This year edition of the conference has received considerable attention from the international community, with 21 (29%) submissions showing at least one author affiliated to a foreign institution. This amounts to a total of 40 authors over 186 (21%) affiliated to 11 foreign countries: Croatia, Czech Republic, France, Germany, Netherlands, Romania, Spain, Sweden, Switzerland, United Kingdom, and United States.

Conference keynotes also offer a view over topics the international community is currently working on, as seen by renowned scientists world-wide. At the 2017 edition we were lucky to have three excellent invited speakers, namely Marco Baroni (Facebook Artificial Intelligence Research, France), Yoav Goldberg (Bar-Ilan University, Israel), and Rada Mihalcea (University of Michigan, USA). Section 3.1 offers a brief summary of their contributions.

As an additional insight over state-of-the-art international research, and under the suggestion of AILC, CLiC-it 2017 newly introduced a call for Research Communications: authors of articles published in 2017 at outstanding international venues in computational linguistics were encouraged to submit short abstracts of their work to be presented orally at the conference. In Section 3.2 we further elaborate on this initiative and on its outcome at the 2017 edition.

3.1 Keynotes

Marco Baroni's presentation, titled "Spectacular successes and failures of recurrent neural networks applied to language", touched upon deep learning methods and more specifically on the power and limitations of Recurrent Neural Networks (RNNs) when representing linguistic knowledge. Specifically, Marco showed us how he and his colleagues tried to probe the syntactic abilities of RNNs even in absence of meaningful lexical information. In other words: is an RNN getting some grammatical judgments correctly because it relies on lexical information which is naturally intrinsic in a given sentence, or is it really detecting grammaticality per se? Marco illustrated experiments where specific constructions in four different languages (Italian, English, Hebrew, Russian) were tested for the prediction of long-distance number agreement. Nonce examples of such constructions were created so as to deprive the RNN of natural lexical information to see if it could rely on grammatical clues only.

Results above strong baselines in all tested languages indicate that it is indeed likely that *the RNN is learning abstract grammatical representations from linguistic input* (Gulordava et al. 2018). Surprisingly, though, the RNN doesn't seem as skilled when probed on apparently simpler tasks to do with *systematic compositionality*. For example, in one

experiment we saw an RNN trained on phrases containing various commands featuring expressions like "run", "walk", "turn left", "turn right", "run twice", "turn left and run opposite thrice", "walk after run", but including only a small set composed of "jump" commands ("jump", "jump left", "run and jump", "jump around twice"). The system was tested on all the remaining "jump" commands (jump twice, jump left and run opposite thrice, walk after jump), and results indicate that it wasn't able to generalise, having failed to learn compositionality aspects.

Yoav Goldberg's presentation, titled "Doing Stuff with Long Short Term Memory Networks", also focused on deep learning methods for sequence processing, again considering RNNs and specialized versions of these models such as long short term memory (LSTM) networks, which use gating mechanisms. A broad range of tasks in natural language processing have been discussed on which Yoav and his collaborators have been able to achieve state of the art results. More specifically, Yoav has discussed a special attention mechanism used for bidirectional LSTM, that has achieved outstanding results on dependency parsing. Other tasks that have been discussed are coordination boundary prediction, morphological reinflection, preposition sense disambiguation, text generation, and machine translation. Using Yoav's own words, LSTM are very capable learners achieving strong results, making reviewers happy, and resulting in the publication of many papers.

Yoav's presentation also touched upon more theoretical issues, relating RNN models to both linguistic representation and formal language theory. Viewing RNNs as trainable functions from vectors to a single vector, one idea is to inspect what information is encoded in the produced (continuous) vectors. Going through several experiments, Yoav has shown that LSTM are capable of encoding word order information from sentences, as well as sentence lengths. Yoav has also asked the question of what kind of syntactic patterns can be represented by means of RNNs. Partially related to work presented by Marco Baroni, as discussed above, in (Linzen, Dupoux, and Goldberg 2016) Yoav has shown that agreement can be learned remarkably well in simple cases, without the need of supervision. However, in the presence of hierarchical syntactic constructions such as those obtained with the use of relative clauses, there is a performance degradation in learning of agreement dependencies, and some sort of supervision is required. Finally, Yoav has asked the general question of what LSTM models are capable of learning, in relation to formal devices such as finite state automata and grammars providing hierarchical representations. The adopted methodology for this investigation involves inspection of vector representations of sentences, as before, as well as mapping of RNN states into discrete states, forming a finite automaton abstraction. Yoav has shown through several experiments that RNN are capable of capturing regular patterns and, up to some extent, also self-embedding patterns typical of hierarchical structures. However, in many cases the representation captured by the RNN is much more complex than the actual concept class being learned.

Rada Mihalcea held a talk titled "*Computational Sociolinguistics - An Emerging Partnership*" focused on the interplay and mutual benefit between computational linguistics and social sciences. Achievements of the former are currently the trigger for several studies on community phenomena as emerging from social networks, such as the analysis of demographic information as well as the recognition of social trends and personal traits. On the social sciences side, specific tasks and problems stimulated attention on new phenomena where the role of linguistic information is crucial, such as demographic text analysis (Garimella, Banea, and Mihalcea 2017), deception recognition

and grounded emotions. In the talk, Rada covered the different topics by surveying latest results on each one.

On the level of demographic text analysis, the talk discussed the recognition of variations of word associations, as a way to characterise communities at the gender or geographical level. Word associations are crucial signals of the mental model behind conceptual connections in the human mind. These are important to characterise the ways humans, since their young age, develop core components of their semantic knowledge. Moreover, *demographic-aware word association models* are a strong basis for demographic-aware NLP: while community specific word similarity or text similarity models are relatively close tasks, future stages of this research may well incorporate demographic-aware labelled associations and keyword extraction, useful for advanced information retrieval, as well as personalised dialogue. The keynote talk presented several results: word associations do vary in fact across user communties, and automatic discovery methods are able to derive the same patterns as those elicited during traditional classroom surveys. Finally, demographic-aware models, based on a skip-gram architecture, are shown to outperfom user agnostic models (Garimella, Banea, and Mihalcea 2017). A second task discussed in the talk was *deception recognition*. The discussion focused on the role of machine learning and on the impact of a variety of features for the modeling of the deception in open sources as well as in specific settings, such as the multimodal deception detection in real-life situations. The talk suggested that detection can be triggered also against short texts according to simple linguistic features, such as bag of words or bigrams. Although verbal information provides evidence on which the agreement among humans is highest, multimodality is highly beneficial (improvements over 10-15% increase in accuracy). Linguistic information in fact fruitfully combines with non verbal features, such as facial displays (e.g. eyebrow or lips) or hand gestures (e.g head movements and trajectories). Needless to stress that gender and age prediction in deceptive texts is still a challenging task.

3.2 Research Communications

As already mentioned, articles published in year 2017 at major CL/NLP conferences and journals could be orally presented within a dedicated session at the conference, called Research Communications, in order to enforce dissemination of excellence in research. This was mainly thought bearing in mind bachelor and master students who do not often get the opportunity to travel to major events until later in their career, and could therefore get a glimpse of international research carried out at Italian institutions. Furthermore, because CLiC-it requires the submitted papers to be yet unpublished, such research would not have made its way to the conference towards the standard submissions channels, but we still deemed it important that it'd be presented and discussed.[2]

Out of 7 submissions for the Research Communications special track, 5 excellent works were selected and orally presented. These had previously appeared at major 2017 conferences, such as ACL (Croce et al. 2017), EMNLP (Basile and Tamburini 2017), and EACL (Karoui et al. 2017), or had just been published in relevant journals, namely Computational Linguistics (Tripodi and Pelillo 2017) and the ACM Transactions on Interactive Intelligent Systems (Zanzotto and Ferrone 2017).

2 Research communications are not published in the conference proceedings.

4. Students and Junior Researchers

AILC's and CLiC-it's attention to students did not end with having a rich, international, technical programme that young researchers can benefit from. Other initiatives have been put in place with students in mind. Three of them were new at the 2017 edition, while one is by now an established tradition at CLiC-it conferences. The three innovations are the following. First, we organised two tutorials aimed at covering both linguistics and as well more computational aspects, given by international experts in the field (Section 4.1). Second, a panel completely dedicated to the discussion of teaching computational linguistics subjects and programmes in Italy, both at the bachelor and master level, with an eye to Europe (Section 4.2). Third, we introduced a prize for the best thesis in computational linguistics defended in the previous year at any Italian university (Section 4.3). While the panel definitely set and opened a discussion that we believe it will be ongoing in the community but not necessarily represented at future CLiC-it conferences in such a form, we do hope that the tutorials and the prize will become a core part of the annual AILC conferences from this edition onwards.

Finally, as in previous editions of the conference, papers featuring a young researcher as first author could be nominated for the *Young Researcher Best Paper Award* (Section 4.4).

4.1 Tutorials

For the first time in the history of the event, CLiC-it 2017 featured two tutorials, one at the beginning and one at the end of the conference.

The first tutorial, titled "Stretching the Meaning of Words: Inputs for Lexical Resources and Lexical Semantic Models", was provided by Elisabetta Ježek, University of Pavia.[3] This tutorial targeted those researchers in CL/NLP who might be less accustomed to lexical theories. It provided an overview of the main properties of words and a description of the structure of the lexicon in terms of word types, word classes, and word relations. The tutorial also introduced the categories that are needed to classify the types of meaning variation that words display in composition, and examined the interconnection between these variations with syntax, cognition and pragmatics.

The second tutorial, titled "Implementing dynamic neural networks for language with DyNet", was provided by Yoav Goldberg, Bar Ilan University.[4] The tutorial targeted those researchers who want to catch up with state-of-the-art neural approaches, with an applied flavour. Several software libraries are available for programming neural network models, such as Theano, TensorFlow and Keras, which assume a fixed (static) graph structure. This tutorial introduced a radically different approach, the DyNet neural networks package, in which the graphs are dynamic and constructed from scratch for every training example. This makes it very easy to program complex networks with structure that depends on the input. In contrast to existing software, which is tailored for the GPU, the DyNet package also works very well on the CPU.

To particularly highlight the importance that such opportunities have for young researchers, all registered students were allowed to freely attend the tutorials. This was

3 http://sag.art.uniroma2.it/clic2017/Jezek2017Clic-itTutorialRome.pdf
4 http://sag.art.uniroma2.it/clic2017/it/2018/01/04/yoav-golbergs-tutorial-materials/

also made possible thanks to the availability of Elisabetta and Yoav, to whom the CLiC-it 2017 chairs as well as the AILC steering committee are particularly grateful.

4.2 Teaching Computational Linguistics and Natural Language Processing in Italy

Computational Linguistics and Natural Language Processing in Italy are usually not taught as dedicated programmes, with the notable exception of the programme in Digital Humanities (Informatica Umanistica) at the University of Pisa. However, as a community, we do believe this should change in the future as the field deserves a proper, clear position in the Italian higher education sphere.

To make things better we need to first know where we stand. Thus, to glean a picture of the current state of things at Italian Universities, we devised two joint initiatives, as the basis for future developments. First, in Spring 2017 we launched a questionnaire aimed at collecting information over all courses taught on Computational Linguistics and Natural Language Processing both at the bachelor and master levels in Italy[5]. A snapshot of the questions asked in the survey (in Italian) is shown in Figure 1. Respondents were asked to fill one questionnaire per course taught.

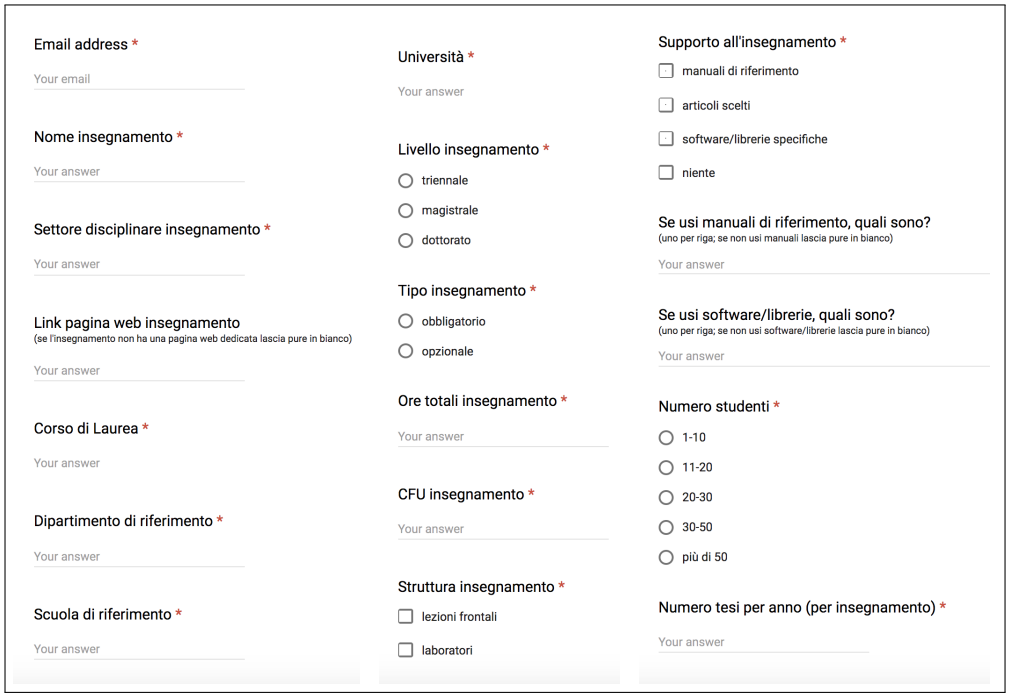

Figure 1
Snapshot of the questionnaire launched in spring 2017 to survey the status of computational linguistics and natural language processing in Italy modules at Italian Universities (`https://goo.gl/9cLzTR`).

Second, we organised a panel at CLiC-it 2017 completely dedicated to teaching. The aim of the panel was to initiate a reflection on the state of things, also discussing the results of the survey. The panel's composition was conceived so as to reflect the

5 `https://goo.gl/9cLzTR`

intrinsic interdisciplinary character of our field, and the fact that courses in computational linguistics are taught within very different programmes. Indeed, we invited a representative of teaching CL/NLP within a humanities programme, a representative of teaching CL/NLP within a science/engineering programme, a representative of teaching CL/NLP within the specific, hybrid Digital Humanities programme in Pisa, and a representative of a gateway to Europe, in the form of the LCT Erasmus Mundus Programme in Language and Communication Technologies. Panelists were therefore as follows:

- Raffaella Bernardi, Università di Trento, Coordinator of the Language and Multimodal Interaction track and local contact of the Erasmus Mundus European Programme in Language and Communication Technologies;
- Alessandro Lenci, Università di Pisa, President of the Degree Programme in Digital Humanities;
- Giovanni Semeraro, Università di Bari "Aldo Moro", Department of Computer Science;
- Fabio Tamburini, Università di Bologna, Department of Classics and Italian Studies (FICLIT).

Each panelist provided a brief overview of their teaching situation and experience, and all together discussed the results of the survey.

At the time of CLiC-it 2017 (December 2017) we had 16 respondents for a total of 26 different courses. The picture that emerges geographically does not accurately reflect the actual situation in Italy, as not everyone responded. Overall, though, the information that we gathered regarding the specifics of the courses is likely to generalise well even to the courses for which no information was provided.

The official areas the courses are taught in are *Linguistics* (L-LIN/01), *Computer engineering* (ING-INF/05), *Computer science* (INF/01), and *Teaching Modern Languages* (L-LIN/02), plus a cross-sector module for PhD students at the University of Trento. The large majority of courses are taught at the master level (77%), and most of the courses overall are optional (61%).

Regarding materials used, there is a wide variety in both fields (humanities and science), with very little overlap. The most used textbook in humanities classes is *Testo e Computer*, by Lenci, Montemagni, Pirrelli (Lenci, Montemagni, and Pirrelli 2005), while the most used in science modules is *Speech and Language Processing*, by Jurafsky and Martin (Jurafsky and Martin 2009). The latter is also one of the only two volumes used in both humanities and science, the other one being *Natural Language Understanding*, by James Allen (Allen 1995). As for tools, those that appear used in both humanities and science modules are OpeNLP, the Stanford Tools, and NLTK, but there is a large number of tools that are only used in computer science modules, and a substantial number of tools used in humanities only. Examples of the former are deep learning modelling frameworks such as Keras, Theano, and Tensorflow, while examples of the latter are more front-end analysis- and annotation-related tools such as the Sketch Engine, the Mate Tools, and Antconc, though we also see tools for morphosyntactic processing, such as TreeTagger and the Malt Parser, in addition to the previously mentioned ones.

The survey and the panel are only just the beginning of an investigation into the Italian teaching context, and while they did open up a reflection and a discussion on the situation of CL/NLP teaching in Italy, the topic will need further and continuous attention. Overall, we believe there is a general consensus in our community over the

need to make our field more independent and more widely recognised as a fully-fledged discipline in the Italian higher education system. Working as a community towards making teaching more homogeneous and more systematically organised is a first step in this direction. The survey, the results and some additional materials are accessible and regularly updated at `https://goo.gl/NZ64Xn`.

4.3 AILC Master Thesis Prize

One more activity intended to recognise excellence in student research was the newly introduced prize for the best Master Thesis (Laurea Magistrale) in Computational Linguistics. This special prize is endorsed by AILC. For this first edition, the committee was composed by a member of the AILC board (Felice Dell'Orletta), a chair of CLiC-it 2016 (Anna Corazza), and a chair of CLiC-it 2017 (Malvina Nissim). Theses defended between January 1st 2016 and July 31st 2017 at any Italian University were eligible for the 2017 prize.

Ten theses were submitted, with the following geographical distribution: Pisa (4),Turin (3), Parma (1), Siena (1), Trento (1). Gender was balanced, with five theses written by female students and five by male students. The evaluation was performed by the three committee members individually in a first stage, after having agreed on a set of specific criteria which had to do both with content (including originality and timeliness of the topic), as well as writing (including clarity, style, and the structure of the thesis). At a second stage, the committee jointly discussed each thesis in details during several Skype meetings, and came up with a short list of three theses, which all deserved the prize. The choice of a final winner was not at all easy, and the reason why eventually we selected the one we selected is its being the closest to the core of our discipline. The first AILC prize for the best master thesis in computational linguistics was thus awarded to:

> Alessio Miaschi, 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"

This is a work that involves both the development of a working system that models a specific language phenomenon, as well as a thorough linguistic analysis based on the features used and on detailed error analysis. All this on top of an excellent background overview, and a view to concrete, future applications, directly useful to society.

The other two theses which made it to the final selection were the following:

> Chiara Alzetta, Università di Pisa: "Studio linguistico-computazionale per l'analisi dei tipi linguistici. Similarità e differenze nel confronto fra Universal Dependencies Treebanks"
>
> Enrico Mensa, Università di Torino: "Design and implementation of a methodology for the alignment of semantic resources and the automatic population of Conceptual Spaces"

As part of the prize, Alessio Miaschi received a monetary sum from AILC, free membership to the association for one year, and free attendance to CLiC-it 2017. At the conference the whole community got the chance to listen to Alessio's presentation of his thesis, right at the end of the panel specifically dedicated to the teaching of computational linguistics and Natural Language Processing in Italy. This was a nice

fit, since the high quality of the submitted works really goes to show how much talent, both among students and among teachers, there is at Italian institutions in the field of computational linguistics.

4.4 Young researcher best paper award

In line with previous editions of the conference, CLiC-it 2017 also featured a best paper award, specially directed to PhD students and young researchers. As a short list, the following papers were initially selected by the Program Committee co-chairs on the basis of the review scores and of the above requirement on young authors: "AHyDA: Automatic Hypernym Detection with Feature Augmentation", by Ludovica Pannitto, Lavinia Salicchi and Alessandro Lenci, University of Pisa; "Deep Learning for Automatic Image Captioning in Poor Training Conditions", by Caterina Masotti, Danilo Croce and Roberto Basili, University of Rome Tor Vergata; and "Deep-learning the Ropes: Modeling Idiomaticity with Neural Networks", by Yuri Bizzoni at Göteborg University, Marco Senaldi and Alessandro Lenci at University of Pisa.

In a second phase, a dedicated jury of five people scrutinized and compared the above papers, and took a final decision to assign the award to the paper

> "AHyDA: Automatic Hypernym Detection with Feature Augmentation", by Ludovica Pannitto, Lavinia Salicchi and Alessandro Lenci, University of Pisa

The awarded paper was presented in the area Semantics and Knowledge Acquisition, and reports experiments on a new method of hypernym detection based on a smoothed version of the distributional inclusion hypothesis (Pannitto, Salicchi, and Lenci 2017).

5. Outreach

At its fourth edition, with the fifth one in preparation, CLiC-it has been the prime forum for researchers in Computational Linguistics in Italy. However, the conference also strives to be a platform for discussion on CL/NLP topics also *outside* the research community. We discuss in this section how CLiC-it 2017 has indeed successfully served as a meeting point for researchers, industry, and the public administration.

As Chairs, we organised one panel which revolved uniquely around the work that was done during 2017 by AGID (Agenzia Italia Digitale) through the creation of a dedicated task force on Artificial Intelligence. Launched by the Italian government, this task force has a specific focus on social challenges, opportunities and perspective regulations of AI. The invited panelists were members of the AI task force, including prof. Giuseppe Attardi, Guido Vetere and Enzo Maria Le Fevre, the person responsible for the task force's communication activities. The panel was moderated by Bernardo Magnini, the president of AILC. The discussion has reaffirmed the important role played by Natural Language technologies in the development of the entire AI field. The strong international grounding of Italian research on CL/NLP confirms the potential of the CLiC-it community as a key actor in the task force's activities.

As already mentioned, the conference has received several submissions from private companies, attesting the growing interest for the field also outside of academia. Beside Facebook and other international organisations such as the European Language Resources Association (ELRA), the support to CLiC-it 2018 mainly came from small and medium Italian enterprises, whose core activity is technological innovation, in particular language and voice technologies. The interesting aspect is that most of them

presented also applied research work on specific and somehow innovative problems. Examples in the technical programme are papers on applications in the area of the Italian Public Administration, the adoption of pragmatics cues to improve dialogue abilities in chatbots, the industrial applications of community question answering methods as well as the automatic evaluation of employee satisfaction through the use of written texts and questionnaires. Along with the obvious application-oriented side effects corresponding to effective methods for original applications, these papers confirm the fruitful cross-fertilisation between industrial topics or challenges and the novel paradigms or techniques emerging from academic research.

Finally, another AILC initiative aimed not only at the promotion and development of tools and resources for Italian NLP, but also at collaboration with industry—both in terms of research as well as in terms of end users—is EVALITA, the Campaign for the Evaluation of NLP and speech tools for Italian (`www.evalita.it`). EVALITA is co-located with CLiC-it, but it's a bi-annual event, and was not scheduled for 2017. Next edition will be co-located with CLiC-it 2018. Nevertheless, several papers presented at CLiC-it 2017 made use of the data produced in the context of the 2016 edition of the campaign (Basile et al. 2017), which goes to show the benefits of creating re-usable resources which become benchmarks for a variety of tasks. We are thus very much looking forward to next EVALITA (`www.evalita.it/2018`) and obviously to next CLiC-it in Turin (`clic2018.di.unito.it/it/home`).

Acknowledgments

Even if CLiC-it is a medium size conference, pulling together the meeting requires major efforts on the part of many people. The Program Committee co-chairs would therefore like to take the opportunity to acknowledge here all of the people that have been involved in the event organization. This conference would not have been possible without the dedication, devotion and hard work of the members of the Local Organising Committee, who volunteered their time and energies to contribute to the success of the event. We are also extremely grateful to the Programme Committee members for producing 207 detailed and insightful reviews, as well as to the Area Chairs who assisted us in many ways. We also want to acknowledge the support from endorsing organisations and institutions and from all of the sponsors, who generously provided funds and services that have been crucial for the realisation of the event. Special thanks are also due to the University of Rome "Tor Vergata" and to the National Research Council of Italy for their support in the organisation of the event and for hosting the conference. Finally, we want to acknowledge the EasyChair infrastructure for the management of the review process and the support in the collection of the camera-ready papers for the composition of the conference proceedings.

References

Allen, James. 1995. *Natural Language Understanding (2Nd Ed.)*. Benjamin-Cummings Publishing Co., Inc., Redwood City, CA, USA.

Basile, Ivano and Fabio Tamburini. 2017. Towards quantum language models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*, Copenhagen, Denmark, September 7-11.

Basile, Pierpaolo, Malvina Nissim, Viviana Patti, Rachele Sprugnoli, and Francesco Cutugno. 2017. EVALITA Goes Social: Tasks, Data. *Italian Journal of Computational Linguistics*, 3(1):93–127.

Croce, Danilo, Simone Filice, Giuseppe Castellucci, and Roberto Basili. 2017. Deep learning in semantic kernel spaces. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 345–354, Vancouver,

Canada, July 30 - August 4.
Garimella, Aparna, Carmen Banea, and Rada Mihalcea. 2017. Demographic-aware word associations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, (EMNLP 2017)*, pages 2285–2295, Copenhagen, Denmark, September 9-11.
Gulordava, Kristina, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless green recurrent networks dream hierarchically. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1195–1205, New Orleans, Louisiana (USA), June 1-6. Association for Computational Linguistics.
Jurafsky, Daniel and James H. Martin. 2009. *Speech and Language Processing (2Nd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.
Karoui, Jihen, Farah Benamara, Véronique Moriceau, Viviana Patti, Cristina Bosco, and Nathalie Aussenac-Gilles. 2017. Exploring the impact of pragmatic phenomena on irony detection in tweets: A multilingual corpus study. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. Association for Computational Linguistics (ACL).
Lenci, Alessandro, Simonetta Montemagni, and Vito Pirrelli. 2005. *Testo e computer: elementi di linguistica computazionale*. Studi superiori. Carocci.
Linzen, Tal, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.
Pannitto, Ludovica, Lavinia Salicchi, and Alessandro Lenci. 2017. Ahyda: Automatic hypernym detection with feature augmentation. In *CLiC-it*, volume 2006 of *CEUR Workshop Proceedings*, Roma, December 11-13. CEUR-WS.org.
Tripodi, Rocco and Marcello Pelillo. 2017. A game-theoretic approach to word sense disambiguation. *Computational Linguistics*, 43(1):31–70.
Zanzotto, Fabio Massimo and Lorenzo Ferrone. 2017. Have you lost the thread? Discovering ongoing conversations in scattered dialog blocks. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 7(2):9.