Multi-Task Sequence Prediction For Tunisian Arabizi Multi-Level Annotation

Elisa Gugliotta[1,2,3], **Marco Dinarelli**[1], **Olivier Kraif**[2]
1. Université Grenoble Alpes - Laboratoire LIG, Getalp group.
2. Université Grenoble Alpes - Laboratoire LIDILEM.
3. Sapienza University of Rome.
elisa.gugliotta@uniroma1.it
{marco.dinarelli,olivier.kraif}@univ-grenoble-alpes.fr

Abstract

In this paper we propose a multi-task sequence prediction system, based on recurrent neural networks and used to annotate on multiple levels an Arabizi Tunisian corpus. The annotation performed are text classification, tokenization, PoS tagging and encoding of Tunisian Arabizi into CODA* Arabic orthography. The system is learned to predict all the annotation levels in cascade, starting from Arabizi input. We evaluate the system on the TIGER German corpus, suitably converting data to have a multi-task problem, in order to show the effectiveness of our neural architecture. We show also how we used the system in order to annotate a Tunisian Arabizi corpus, which has been afterwards manually corrected and used to further evaluate sequence models on Tunisian data. Our system is developed for the Fairseq framework, which allows for a fast and easy use for any other sequence prediction problem.

1 Introduction

In the last decade neural networks became the state-of-the-art models in most NLP problems. Sequence-to-sequence models (Sutskever et al., 2014; Vaswani et al., 2017), built on top of recurrent (Hochreiter and Schmidhuber, 1997; Cho et al., 2014), convolutional (Gehring et al., 2017; Wu et al., 2019) or attentional (Bahdanau et al., 2014; Vaswani et al., 2017) modules, and structured in encoder-decoder architectures, are currently the most effective models for NLP problems. Neural networks have been used also for multi-task learning since early in their diffusion (Collobert and Weston, 2007; Collobert and Weston, 2008; Collobert et al., 2011).

As a semitic language, Arabic has a highly inflectional and derivational morphology, which makes Arabic processing an engaging challenge. This morphological complexity has traditionally been handled through morphological analysers, such as BAMA (Buckwalter, 2004), which has been used by the Linguistic Data Consortium (LDC) to develop the Penn Arabic Treebank (PATB) (Maamouri et al., 2004). Recently, the number of NLP contributions to morphological analysis, disambiguation, Part-of-Speech (PoS) tagging and lemmatization has increased substantially, for both Modern Standard and Dialectal Arabic (MSA and DA, respectively). Multitask learning was proved to be an effective way to process Arabic morphology for MSA fine-grained PoS tagging (Inoue et al., 2017), as well as for DA (Zalmout and Habash, 2019). Concerning NLP applied to DA, it is possible to observe two main macro-strategies aimed at remedying the lack of data for DA: **1.** *MSA systems adaptation to DA processing*, like (David et al., 2006) who exploited the Penn Arabic Treebank (PATB) (Maamouri et al., 2004) and used explicit knowledge about the relation between MSA and Levantine Arabic. Instead, (Duh and Kirchhoff, 2005) built a PoS tagger for Egyptian through a minimally supervised approach by leveraging the CallHome Egyptian Colloquial Arabic corpus (ECA). **2.** *The constitution of new resources not based on MSA-DA relations*, in particular dialectal corpora, such as the Fisher Levantine Arabic Conversational Telephone Speech (Maamouri et al., 2007).[1] This second strategy has been followed also collecting more *ad-hoc* resources. (Bouamor et al., 2018) presented the first parallel DA corpus, collecting the dialects of 25

[1]These resources are not freely available.

Proceedings of the Fifth Arabic Natural Language Processing Workshop, pages 178–191
Barcelona, Spain (Online), December 12, 2020

Arab cities, including the Tunisian dialects of Tunis and Sfax. The MADAR corpus has been created by translating selected sentences from the Basic Traveling Expression Corpus (BTEC) (Takezawa et al., 2007). Regarding Tunisian Dialect (TD), the resource constitution strategy has been instantiated as MSA resource adaptation to the DA, e.g. building lexicons (Boujelbane et al., 2013), PoS-taggers (Boujelbane et al., 2014; Hamdi et al., 2015), morphological analysers (Zribi et al., 2013) or morphological systems to disambiguate annotated transcriptions (Zribi et al., 2017). Considering the lack of freely available resources, we opted for an approach similar to the one used in *Curras* Palestinian corpus collection (Jarrar et al., 2017), which exploits MADAMIRA tools (Pasha et al., 2014), (cf. section 4.1).

The development of informal online communication provided a solution to most of the data availability problems, making accessible to the scientific community a large amount of texts, both written and oral. Concerning texts written in DA, it is possible to find two main writing systems: Arabic and Latin scripts. With regard to the second one, letters are used together with digits for the encoding of those Arabic letters without correspondence in the Roman alphabet. This system is already well known as *Arabizi*, or *Arabish* for non-Arabic speakers. Most of the work developed on Arabish focus on language identification (Darwish, 2014) and sentiment analysis (Duwairi et al., 2016; Fourati et al., 2020). Several works are focused on the conversion of Arabish into Arabic script, as the Parallel Annotated Egyptian Arabish-Arabic Script SMS/Chat Corpus (Bies et al., 2014). Transliteration has also been addressed for Tunisian Arabish (Masmoudi et al., 2015; Masmoudi et al., 2019; Younes et al., 2020).

In this paper we propose a multi-task sequence prediction system based on recurrent neural networks, that we used to annotate at multiple levels the Tunisian Arabish Corpus (TArC) (Gugliotta and Dinarelli, 2020). The annotation levels include tokenization, Part-of-Speech (PoS) tagging and Tunisian Arabish encoding into Arabic script. The system is learned to predict all the annotation levels in cascade, starting from Arabish input. We evaluate the system on the TIGER German corpus (Brants et al., 2004) in order to show the effectiveness of our neural architecture. While the purpose of this evaluation is not to improve state-of-the-art on this task, our results are comparable and sometimes better than the best published models. We show also how we used the system in order to annotate TArC, which has been afterwards manually corrected and used to further evaluate sequence models on Tunisian data. Our system is developped for *Fairseq*[2] (Ott et al., 2019), it can therefore be used for any problem involving sequence prediction.[3]

In the remainder of the paper we describe the TArC corpus, that we annotated with multi-level information, and we used to evaluate our neural system (in section 2). In section 3 we describe our multi-task neural architecture for multi-level annotation, in section 4 we describe the TIGER corpus, the experimental settings, and all the results obtained with our system, on both TIGER and TArC corpora. We conclude the paper in section 5.

2 Tunisian Arabish Multi-Level Annotated Corpus

The corpus used in this paper is the Tunisian Arabish Corpus (TArC) (Gugliotta and Dinarelli, 2020), the result of a multidisciplinary work with a hybrid approach based on: 1. dialectological research questions; 2. corpus linguistics standards and 3. deep learning techniques. TArC has been conceived with the aim to extend the dialectological investigation to the web, not only considering it as a new resource for linguistic analyses, but mainly because the object of TArC is a Computer Mediated Communication (CMC) writing system.

The gathering of CMC corpora for linguistic study purposes is a long-standing practice: as early as the 1990s, in order to study linguistic and communicational aspects, researchers began to collect corpora from mailing lists, newsgroups, electronic conferences or chat rooms (Yates, 1996; Todla, 1999; Berjaoui, 2001; Feldweg et al., 1995). Nowadays, the study of CMCs is a research domain it-self, crossing various disciplines such as sociology and linguistics. The linguistic questions related to CMC-corpora may for example concern paraverbal phenomena and the expression of emotions (Riordan and

[2]https://github.com/pytorch/fairseq

[3]Our system, with data used in this paper, is available at https://gricad-gitlab.univ-grenoble-alpes.fr/dinarelm/tarc-multi-task-system.

The last updated version of TArC is available at https://github.com/eligugliotta/tarc.

Kreuz, 2010; Tantawi and Rosson, 2019), politeness formulas and the degree of message formality (Brysbaert and Lahousse, 2019), the effects of orality in written communication (Soffer, 2010), the role of code-mixing and code-switching in mediated discourse (Morel and Doehler, 2013; Mave et al., 2018), their graphic and orthographic characteristics (Sullivan, 2017) (concerning Arabic). Lastly, a lot of research deals currently with the automatic processing of such corpora (Lopez et al., 2018; Panckhurst, 2017).

Among the purposes of dialectology there is the dialect collection and description with traditional approaches: fieldwork, oral text collection and transcription, glossary building. We observed that in the case of Arabic varieties the descriptive landscape is made of multiple studies on single phenomena. For this reason, we developed a resource inspired by dialectological investigation, which borrows the principles of corpus linguistics in order to guarantee representativeness, accessibility, balance and authenticity of the linguistic data (Szmrecsanyi and Anderwald, 2018; Wynne, 2005). The data gathered in TArC, together with various metadata, takes a snapshot of Tunisian Arabish writing and its evolution over the last ten years. TArC is built selecting data with the following criteria: 1. *text mode*: informal writing; 2. *text genres*: forum, blog, social networks, rap lyrics; 3. *domain*: CMC; 4. *language*: Tunisian; 5. *location*; 6. *publication date*. The last two items were registered via metadata extraction (publication date, user's age, gender and provenience).

The building process automation overcomes the *observer's paradox* problem (Labov, 1972), an issue much discussed in dialectology (Boberg et al., 2018). It also allows the reproducibility of the work, as well as the quantitative extension of an open corpus (such as TArC), which is normally difficult to ensure by dialectological research. TArC collection has therefore been enhanced thanks to the multi-task architecture, used for a semi-automatic annotation (cf. section 3) to get as close as possible to a consistent linguistic annotation (Wynne, 2005). The automatically generated annotations were post-edited by a linguist qualified in Arabic language and Tunisian variety, whose work was occasionally verified by native speakers.[4] Such annotation work complies with both the *applicative* and the *analytical* purposes of a corpus. The former concerns the generation of NLP tools for the Tunisian Arabish processing. The latter is realised through the multi-functional annotation levels of TArC, which allow global and systematic studies of Tunisian variety and its Arabish encoding. This way, TArC usefulness returns to the dialectological area, the field in which the preliminary research questions were addressed.

TArC has been annotated with four information levels. **1) Classification** of words in three classes: *arabizi*, *foreign* and *emotag*. The first class is for Tunisian and MSA words, the second one is to classify non-Arabic code-mixing; the third is used for elements as smileys or emoticons. **2) Encoding in Arabic script** in Conventional Orthography for Dialectal Arabic (CODA*) (Habash et al., 2018). **3) Tokenization**, Tunisian words encoded in CODA* have been tokenized following the *D3_BWFORM* configuration scheme where basically all clitics are tokenized, including the article (Pasha et al., 2014). **4) Part-of-Speech** according to the *PATB* guidelines (Maamouri et al., 2009). All levels have been developed following the same incremental and semi-automatic procedure described in (Gugliotta and Dinarelli, 2020) for the CODAfying stage.

3 Multi-Task Sequence Prediction System

There are several works about multi-task learning with neural networks for NLP problems (Wu and Huang, 2015; Luong et al., 2016), *inter alia*. Most of the time the neural architecture factorises some parameters for information that can be shared among tasks, and then uses different modules (e.g. decoders) for each task, which are learned independently.

As described in section 2, our goal for Tunisian Arabish data is a multi-level annotation scheme, where the different levels are potentially related. From an NLP point of view, this relations imply that some levels of annotation may help disambiguation when annotating other levels. For instance the classification information can disambiguate annotation into CODA*, tokenization and PoS tagging. Intuitively we expected that learning tasks in chain, organised in a cascade manner in a neural network, would benefit to each other, in contrast to learning tasks individually.

[4]Due to COVID-19 lockdown it was not possible to conduct the field research scheduled for March 2020.

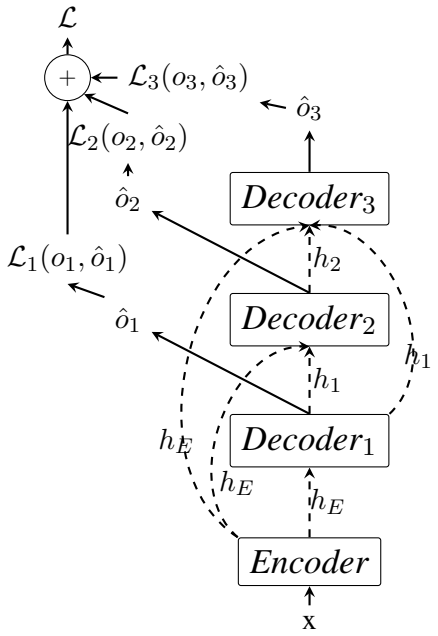

Figure 1: A high-level schema of our multi-task neural system

3.1 Multi-Task Neural Architecture

We follow the intuition above and we propose a multi-task neural architecture where the different learned tasks are organised in a cascade. The input is the Arabish text. The outputs, corresponding to the tasks to be learned, are, in this order, the classification information, the conversion into CODA* orthography, the tokenization of the *CODAfied* tokens and the PoS tags. Outputs from previous tasks are reused by the following tasks, they are thus learned jointly and interdependently. The input is transformed into hidden context-aware representations with an encoder based on recurrent layers. The outputs are processed by different decoders, each of them taking as input the hidden state of the encoder, and the hidden state of each of the previous decoders. The output of each decoder is used to learn each task.

More formally, let the task i be represented by the model $M_i(x, H_i)$, with x the input (Arabish text representations), and H_i the list of hidden states from the previous models, plus the current model's hidden state h_i. Each model i generates an output $\hat{o}_i$ and a hidden state h_i. $\hat{o}_i$ is the predicted output, which is used to learn the task i by computing a loss $\mathcal{L}_i(o_i, \hat{o}_i)$ comparing $\hat{o}_i$ to the expected output o_i. Internally the global model M is made of an *Encoder* and I decoders $Decoder_i$, with $i = 1 \dots I$. The list H_i includes both the encoder hidden state h_E and the decoders hidden states $h_1 ... h_i$. An high-level schema of this architecture, with the flow of information for three tasks ($I = 3$), is presented in figure 1. All the tasks are learned jointly by minimising the global loss $\mathcal{L} = \sum_i \mathcal{L}_i(o_i, \hat{o}_i)$, on top of the circled $+$ in the schema (Figure 1).

Like in the original sequence-to-sequence model based on an attention mechanism (Bahdanau et al., 2014), each decoder attends to encoder and decoder's hidden state information with an attention mechanism. The decoder $Decoder_i$ has therefore i different attention mechanisms, one for attending encoder's information, and one for each previous decoder's hidden state. The queries for the attention mechanisms are always the $Decoder_i$'s hidden states, while keys and values are the encoder and previous decoder hidden states. The attention vectors computed by the attention mechanisms are *simply* summed together to generate the final state, used to predict the next output.[5]

[5]We note that we have been testing also gating mechanisms to *blend* the outputs of the attention mechanisms like in (Miculicich et al., 2018), but this always gave worse results than the sum.

	Training		Dev		Test	
# sentences	40 472		5 000		5 000	
	Words	Labels	Words	Labels	Words	Labels
# tokens	719 530	–	76 704	–	92 004	–
dictionary	77 220	681	15 852	501	20 149	537
OOV%	–	–	30,90	0,01	37,18	0,015

Table 1: Statistics of the German corpus TIGER

	Sentences	Words		
Total	4121	32 062		
		arabizi	*foreign*	*emotag*
forum	756	6039	5856	14
social	3146	11 843	3614	587
blog	219	3763	343	3

Table 2: Statistics of the already annotated part of TArC

4 Evaluation

4.1 Data

In order to evaluate our multi-task system, we used two different corpora. One is TArC, described in section 2, the other is the German TIGER corpus (Brants et al., 2004).

TArC corpus has been initially collected from forums, social media and blogs, for a total of 32 062 words, and recently extended to 43 313 words by adding the text type of rap lyrics. In order to better organise the automatic annotation and the manual-correction stages, we split the initial corpus into blocks of roughly 6 500 tokens. Statistics of TArC are presented in table 2. The initial model, used to bootstrap the corpus annotation, has been trained using 2000 sentences from the Tunisian MADAR corpus. MADAR data are well-formed texts encoded in Arabic script, this avoid any code-switching and spelling inconsistency. We processed MADAR data using the MADAMIRA tool (Pasha et al., 2014).[6], producing tokenization and PoS tags. After a manual correction, we obtained the first TArC training block for starting the annotation procedure.

The German corpus TIGER (Brants et al., 2004) is annotated with rich morpho-syntactic information. These include PoS tags, but also gender, number, cases, and other inflection information, as well as conjugation information for verbs. The combination of all these components constitutes the output labels. We used the same data split used in (Lavergne and Yvon, 2017). Statistics of this corpus are given in table 1.

4.2 Settings

4.2.1 Data Pre-processing

We first describe some data pre-processing performed on both corpora, in order to better exploit the small amount of data in TArC, on one side; on the other side, we performed a similar pre-processing on the TIGER corpus, in order to have similar experimental settings and therefore be able to validate the multi-task model with results comparable with the literature.

The TIGER corpus has been used as a benchmark for our multi-task system, before applying it to TArC. Since TIGER data are not natively multi-task, we re-organised TIGER labels in two parts: the first consisting of the PoS core-tag only, the second consisting of the whole label. For example, given the label *ADJA.PoS.Nom.Sg.Masc*[7], we take the PoS tag *ADJA* as a first level of information, and the whole label as a second level. This simple pre-processing allows to have two tasks to learn with our

[6]Version used: MADAMIRA 2.0. D3_BW* schemes (Habash, 2010).

[7]The different pieces stand for adjective, possessive, nominative, singular and male, respectively.

system: a coarse and a fine-grained morpho-syntactic tagging, where the second task, more complex, can be learned using also the information of the first, which is simpler.

In order to reduce data sparsity in TArC, we performed sequence prediction at each annotation level using sub-token units, except for the classification level. Sub-token units are characters for Arabish, *CODAfied* tokens and tokenization levels. For the PoS tags we performed an *ad-hoc* split into coarser units. The PoS tags annotated in TArC follow the LDC guidelines described in (Maamouri et al., 2009).[8] The tags contain rich information, like for the TIGER labels, describing the morphological structure of tokens. For instance the tag *PV-PVSUFF_SUBJ:3MS+[PREP+PRON_2S]PVSUFF_IO:2S*, contains information about a verb with inflectional morphology (*PV-PVSUFF_SUBJ:3MS*), plus information on a pre-pronominal enclitic group attached to the verb (*PREP+PRON_2S*). This group contains also an indirect object in suffix form (*PVSUFF_IO:2S*). Each of these 3 macro components contains person features, *3MS* for the verb, *2S* for the enclitic pronoun, and *2S* for the indirect object suffix.

Quite intuitively, such complex tags, taken as a whole, are very rare in the data. Indeed more than half of them occur only once in our data.[9] However their components are quite common (e.g. *PV, PVSUFF, SUBJ, 3, M, S* and so on). For this reason we split the tag above into a sequence of components like: *PV, PVSUFF, _SUBJ, :3, @M, @S, +, [, PREP, +, PRON, _2, @S,], PVSUFF, _IO, :2, @S*. Symbols like @ are added for the post-processing phase to correctly reconstruct the whole tag. For the same reason, each time a tag is split in this way, the components are wrapped with start and end markers *¡SOT¿, ¡EOT¿* (for Start and End Of Token). A whole tag sequence, associated to an input sentence, is made by concatenating the sequences resulting from the split of each tag. The same start and end markers are used also for the other annotation levels, which are split into single characters, so that the model can learn itself that each token in the input sequence corresponds to one token in all the annotation levels.

In order to have the same settings for TArC and TIGER data, we split the input tokens in the TIGER data into characters, adding the start and end markers. The labels are left unchanged, beyond artificially creating 2 label levels to test our multi-task system (we actually performed experiments also splitting TIGER labels into components, cf. table 3 and 4).

The TArC classification level was added first. This was done using a character-level model pre-trained exploiting: i) the Hussem Ben Belgacem's French dictionary, consisting of 336,531 tokens.[10], and ii) a Tunisian Arabish dictionary of 100,936 tokens, resulting from the merge of the TUNIZI Sentiment Analysis Tunisian Arabic Dataset (Fourati et al., 2020)[11] and the TLD dataset (Younes et al., 2015).

In order to obtain an *emotag* dictonary, we extracted all the smileys and emoticons from the Arabish dictionary above. Once the model was pre-trained on the above data, it was possible to apply also to this annotation level the semi-automatic and incremental annotation procedure used in (Gugliotta and Dinarelli, 2020). At the end of the procedure, the model reached 97% of accuracy. All data were manually checked and corrected.

4.2.2 Model Settings

Concerning model settings, we note that encoder and decoders in our multi-task neural models are all LSTM (Hochreiter and Schmidhuber, 1997).[12]

An optimisation of hyper-parameters like learning rate, dropout ratio (Srivastava et al., 2014), layer size, etc. has been performed on development data of TArC. For experiments on TIGER the same hyper-parameters have been used. The goal here is not to obtain the best absolute results on this task, it is to show that our system is competitive enough to be used safely on unpublished data. Such hyper-parameter optimal values resulted in: $5E^{-4}$ for learning rate, 0.5 for dropout ratio (at all layers, including embeddings), 5.0 for gradient clipping (Pascanu et al., 2012), 256 for both embeddings and hidden layer size (for all layers). We share all embeddings, at input and output layers, and in encoder and decoders.

[8] In the concatenation style we used "-" and the square brackets, to better manage the information through our model.

[9] More precisely, 423 PoS tags out of 776 in the dictionary, that is 54.9%, occur only once.

[10] https://github.com/hbenbel/French-Dictionary (last access on 15/09/2020).

[11] https://github.com/chaymafourati/TUNIZI-Sentiment-Analysis-Tunisian-Arabizi-Dataset (last access on 15/09/2020).

[12] The system is however generic, and potentially any kind of encoder and decoder available in Fairseq may be used. We are currently working on adding the use of Transformer encoder and decoders (Vaswani et al., 2017).

The loss functions used in all our experiments, for all the decoder outputs (see $\mathcal{L}$, $\mathcal{L}_1$, etc. in section 3.1), are the cross-entropy loss. All models are learned with an ADAM optimiser (Kingma and Ba, 2014) with default parameters. Model's outputs are evaluated with the accuracy, after applying post-processing to reconstruct original tokens. This means that if a single character or component in a token is wrong, the token is considered wrong in the accuracy.

4.3 Results

We present first results obtained on the corpus TIGER. We remind that we artificially performed multi-tasking on TIGER by isolating the core-tag from its features for each morpho-syntactic tag, and using the core-tag and the whole one as separated output to be predicted (see section 4.2).

The first set of experiments was performed to choose the optimal number of layers in each decoder of our multi-task system. Results are shown in table 3, the two tasks are **PoS**, for core-tags only, and **MORPHO** for core+feature tags. The results of both tasks show that the model performs at best with 3 layers in each decoder, though the gain with respect to the other choices is small. Despite the gain is small, we observed consistently the best results, both in terms of accuracy and loss values, and on both corpora, with 3 layers.

In the table 3 we show also the comparison of our results with the literature. To the best of our knwoledge the best results on the corpus TIGER have been published in (Dinarelli and Grobol, 2019), which improved previous state-of-the-art of (Lavergne and Yvon, 2017). Our results are comparable with the state-of-the-art, even slightly better on morpho-syntactic tagging, *Dev data*. We would like to insist on the fact that experiments on TIGER have been performed not with the goal to improve the state-of-the-art, but only for validating our multi-task system for performing multi-level annotation of TArC as multi-tasking. In this respect, the model used in (Dinarelli and Grobol, 2019) is quite sophisticated, it performs sequence labelling exploiting both token and character information on the input side, and performing bidirectional decoding on the output side. Our model performs decoding at character-level only, though using several layers over 2 tasks. Beyond this comparison, we consider our results on the TIGER corpus satisfactory for a multi-task setting.

The last 3 lines of table 3 and 4 show results on TIGER Dev and Test data, respectively. In these experiments we compare models learned for decoding label components, instead of whole labels, using character-level input (*Char decoding*), models learned with whole tokens on input and output side (*Token decoding*), and models combining both information, but learned from whole-token tag sequences (*Token+char decoding*). As we can see, *Char decoding* setting is by far the most effective. Combining token and character level information largely improves the *Token decoding* setting, but it is still much less effective than the *Char decoding* setting.

It could be interesting to observe which gain can be achieved with a multi-task model, e.g. on PoS tagging, with respect to a mono-task sequence-to-sequence model on the same task. In order to show such gain, we performed an experiment of PoS tagging with our multi-task system in a mono-task setting, with the same experimental settings. We compare this result with the multi-task counter-part in table 3. The two results are shown in table 5. As we can see, a substantial gain can be achieved performing PoS tagging as part of a multi-task setting. Even if, when learned for multi-tasking, PoS tagging is the first task and so it cannot exploit information coming from preceding tasks, the gain is given by the back-propagation of the morpho-syntactic tagging error through the whole network. Once again, results are obtained decoding at character level only for keeping the same experimental settings as for the TArC.

Experiments on TArC are divided in two phases, corresponding to two annotation phases: the first concerns the Arabish conversion into Arabic script. The second phase consists in classification of each token in *arabizi*, *foreign* or *emotag* classes, together with tokenization of Arabic-encoded tokens, and PoS tagging. Each phase was performed with a semi-automatic procedure, where a model was trained on a first block of data. Such model was used to annotate another block of data. This was then manually corrected and added to the training data. A new model was trained and used to annotate a new block. This procedure was iterated up to the annotation of the full corpus (32 062 tokens).

For the first phase of the annotation (Arabic script encoding only) we used the mono-task sequence-

Corpus: TIGER Dev data		
Best results		
	PoS	**MORPHO**
(Dinarelli and Grobol, 2019)	98.37%	93.94%
Our results		
Model	**LSTM**	
Task	**PoS**	**Morpho**
1 Enc + 1 Dec layers	97.83%	93.16%
2 Enc + 2 Dec layers	98.16%	93.58%
3 Enc + 3 Dec layers	98.30%	94.10%
Char decoding	98.30%	94.10%
Token decoding	96.21%	86.89%
Token+char decoding	98.11%	90.70%

Table 3: Summary of results, in terms of accuracy, obtained on the TIGER development data set with the Tarc Multi-Task system.

Corpus: TIGER Test data		
Best results		
	PoS	**MORPHO**
(Dinarelli and Grobol, 2019)	97.74%	91.86%
Our results		
Model	**LSTM**	
Task	**PoS**	**Morpho**
Char decoding	97.44%	91.81%
Token decoding	94.44%	83.37%
Token+char decoding	97.25%	87.87%

Table 4: Summary of results, in terms of accuracy, obtained on the TIGER test data set with the Tarc Multi-Task system.

to-sequence model of (Dinarelli and Grobol, 2019). Indeed the Arabic script encoding of tokens is the most costly and difficult phase, so we thought it could be easier to have it first, annotating the other levels afterwards. The Arabic script encoding accuracy of the model was below 70% for the first block. This still allowed the annotator to correct the block 3 times faster than if the block was annotated from scratch. For the following data blocks, accuracy of the model increased progressively, up to roughly 76% for the fourth block. At this point we started the second phase, which included the annotation of the fifth and last block with encoding conversion.

In the second phase, we repeated the iterative semi-automatic annotation procedure of the first phase for the classification, tokenization and PoS tagging levels. These were performed with the multi-task system. The first model for bootstrapping the annotation procedure was trained on a part of the MADAR data (Bouamor et al., 2018) consisting of roughly 12,000 tokens (2,000 sentences). These data were annotated with tokenization and PoS information using MADAMIRA as explained in section 4.1, and then manually corrected. The classification information was added manually, which was trivial since all tokens belong to the *arabizi* class in this data. The model trained on MADAR data has been used to annotate the first block of TArC, which is the step 0 of the iterative procedure. In the following 3 iterations, the MADAR data were used together with the TArC blocks already manually corrected. The input for these 3 steps was thus the *CODAfied* Tunisian. Exploiting MADAR was only possible up to the 4th block, since the blocks after the fourth were not already provided with *CODAfied* tokens (see the 1st annotation phase above). However, we planned to add all the annotation levels to the 5th block, including

Corpus: TIGER Dev data	
PoS tagging results	
Model	**LSTM**
Mono-task	95.66%
Multi-task	98.30%

Table 5: Comparison of results of PoS tags decoding from source characters, on the TIGER development data with mono-task and multi-task models.

the encoding in Arabic script level, with the multi-task system. The 5th block was thus annotated using only TArC four blocks in Arabish as training data. At each iteration step, the Arabish data were split randomly into train and validation (dev) sets, so that the dev set is representative of the whole data at each iteration.[13]

We report the results on the 3 tasks of the first 4 steps, where the input was *CODAfied* Tunisian, and the results on the 4 tasks of the following steps, where the input was Arabish, in table 6. The tasks are indicated in the table with **Class** for classification, **Arabic** for Arabic script encoding, **Token** for tokenization, and **PoS** for PoS tagging, respectively. In the column **"Train. tokens"** of the table we report the number of training tokens for each step. Between parenthesis, when this is meaningful, we also report the number of training tokens coming from TArC (the remainder is from the MADAR corpus).

In table 6, *Step0* is the bootstrapping step, where the model is trained on MADAR data only. Results are on a randomly chosen dev data set consisting of 15% of the whole data set. Starting from *Step1*, the dev data set is a 15% random split of the TArC data only, as we are interested in the effectiveness of our multi-task system on spontaneous and informal writing data for annotation purposes.

Results in table 6 prove that the multi-task system is effective also on TArC, especially taking into account the small amount of data available for training the models. The classification task (**Class**) is quite well solved, as at best the model, when evaluated on TArC text, is over 97% of accuracy. Results for tokenization (**Token**) are also satisfactory, in particular at step 3, where the model is over 91% of accuracy. Results on PoS tagging (**PoS**) are quite lower with respect to the other tasks, but we note that this task is the most difficult, among the 3 of the first 4 steps. Indeed, classification only consists in associating to each token one of the 3 classes *arabizi*, *foreign* or *emotag*. The tokenization task consists in splitting a *CODAfied* token into its components with some orthographical transformations, input and output script is thus the same, the model needs to learn the splitting. In contrast, PoS tagging is a conversion from Arabic characters into PoS components.

As we have explained in section 4.2, PoS tags are quite complex, and splitting them into components allows to mitigate the problem of data sparsity. Moreover, accuracy is computed after post-processing, that is after PoS tags have been reconstructed from components. A single mistake on a component results in a wrong tag, affecting the accuracy. Taking all of that into account, we consider the best PoS tagging result of 76.38% of accuracy as an acceptable result.

In table 6 we observe a substantial drop of results from step 0 (where the model is evaluated on the MADAR dev set) to step 1 (where the model is evaluated on the Arabish dev set only).[14] This is not surprising, as MADAR is made of morphosyntactically well-formed text, while TArC is made of CMC spontaneous texts. This behaviour is useful to explain the difference of results between step 3 and step 4 and 5. Beyond that, the increased amount of TArC data with respect to MADAR data through steps 1 to 3, allows to improve results obtained on the MADAR data (*Step0*).

Results in table 6 drop again between steps 3 and 4. We remind that at step 3, data blocks from 1 to 3, plus the MADAR data, are used for training the model, a 15% split of the TArC data are used for validation, and the model is used to annotate the fourth data block. At step 4 only TArC data are used for training, again a 15% split is used for validation, and the fifth block is annotated. At this step an

[13] In this respect, we note that data in different blocks are heterogeneous, as they are not all from the same source. Hence keeping the same dev data set for all the iterations would not be representative.

[14] All MADAR tokens are classified as *arabizi*, it is thus normal that the model gets almost perfect result in classifying it.

Task	Train. tokens	LSTM			
		Class	**Arabic**	**Token**	**PoS**
Corpus: MADAR					
Step0	12 391	99.83%	-	88.83%	72.71%
Corpus: MADAR+TArC					
Step1	17 261 (4 870)	92.69%	-	77.66%	59.56%
Step2	22 173 (9 780)	97.21%	-	87.53%	74.30%
Step3	27 270 (14 870)	96.69%	-	91.47%	76.38%
Corpus: TArC					
Step4	22 150	96.83%	75.30%	73.38%	69.76%
Step5	27 435	97.17%	75.08%	73.07%	66.24%
$\text{Step4}_{\text{smart-init}}$	22 150	95.91%	76.55%	74.96%	72.57%
$\text{Step5}_{\text{smart-init}}$	27 435	97.08%	77.83%	75.69%	69.76%

Table 6: Summary of results, in terms of accuracy, obtained on the TArC data at the different steps of the iterative procedure for semi-automatic annotation of the corpus. The tasks are indicated with **Class** for classification, **Arabic** for Arabic script encoding, **Token** for tokenization, and **PoS** for PoS tagging.

additional task is performed: encoding of Arabish into CODA*.

As we can see in table 6, all results except for classification, substantially dropped. This is due to having an additional task with respect to the previous steps, and thus an additional decoder in the system, and to the use of a smaller training set. We note however this drop is similar to the one between steps 0 and 1. We conclude thus that MADAR well-formed texts have a positive effect on learning spontaneous Arabish text. It is interesting to observe that the drop in PoS tagging results with respect to tokenization, at steps 4 and 5, is much smaller than the drop at steps 1 and 3. This suggests to improve Arabish *CODAfication* results, which may be achieved by adding Arabish encoding to MADAR. Results on the step 5 are similar to step 4. This is not surprising as well, since data in the block 5 have a different style, coming from a different source (blogs). This balances the increased amount of data for training the model.

In order to exploit the MADAR data also at steps 4 and 5, we designed an *ad-hoc* parameter initialisation using the model trained at step 0. Note that such model has a different architecture as MADAR is in Arabic script, it doesn't contain Arabish.[15] Results obtained with this initialisation are reported in the last lines of table 6 marked as *smart-init*. As we can see, except for the classification task which is biased by the fact that in MADAR all tokens are in the *arabizi* class, all other task results improved with respect to step 4 and 5 without pre-initialisation.

5 Conclusions

We presented a multi-task sequence labeling system based on recurrent neural networks, developed for the Fairseq framework and used to annotate TArC on multiple levels. The annotation levels provided are: classification, tokenization, PoS tagging and encoding of Tunisian Arabish into Arabic script, according to CODA*. We described the annotation procedure, after showing the effectiveness of our neural architecture with an evaluation on the TIGER German corpus. As a next stage we plan to expand TArC quantitatively to improve the results and its usability in linguistics and NLP fields. Future work includes qualitative extension through the addition of further annotation levels, such as lemmatization.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

[15] This resulted in a quite task-specific parameter initialisation.

Nasser Berjaoui. 2001. *Aspects of the Moroccan Arabic orthography with preliminary insights from the Moroccan computer-mediated communication*. na.

Ann Bies, Zhiyi Song, Mohamed Maamouri, Stephen Grimes, Haejoong Lee, Jonathan Wright, Stephanie Strassel, Nizar Habash, Ramy Eskander, and Owen Rambow. 2014. Transliteration of arabizi into arabic orthography: Developing a parallel annotated arabizi-arabic script sms/chat corpus. In *Proceedings of the EMNLP 2014 workshop on Arabic natural language processing (ANLP)*, pages 93–103.

Charles Boberg, John A Nerbonne, and Dominic James Landon Watt. 2018. *The handbook of dialectology*. Wiley Online Library.

Houda Bouamor, Nizar Habash, Mohammad Salameh, Wajdi Zaghouani, Owen Rambow, Dana Abdulrahim, Ossama Obeid, Salam Khalifa, Fadhl Eryani, Alexander Erdmann, et al. 2018. The madar arabic dialect corpus and lexicon. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Rahma Boujelbane, Mariem Ellouze Khemekhem, Siwar BenAyed, and Lamia Hadrich Belguith. 2013. Building bilingual lexicon to create dialect tunisian corpora and adapt language model. In *Proceedings of the Second Workshop on Hybrid Approaches to Translation*, pages 88–93.

Rahma Boujelbane, Mariem Mallek, Mariem Ellouze, and Lamia Hadrich Belguith. 2014. Fine-grained pos tagging of spoken tunisian dialect corpora. In *International Conference on Applications of Natural Language to Data Bases/Information Systems*, pages 59–62. Springer.

Sabine Brants, Stefanie Dipper, Peter Eisenberg, Silvia Hansen-Schirra, Esther Konig, Wolfgang Lezius, Christian Rohrer, George Smith, and Hans Uszkoreit. 2004. TIGER: Linguistic interpretation of a german corpus. *Research on Language and Computation*, 2(4):597–620, dec.

Jorina Brysbaert and Karen Lahousse. 2019. Computer-mediated versus non-computer-mediated corpora of informal french: Differences in politeness and intensification in the expression of contrast by au contraire. *Social Media Corpora for the Humanities (CMC-Corpora2019)*, page 48.

Tim Buckwalter. 2004. Buckwalter arabic morphological analyzer (bama) version 2.0. *Linguistic Data Consortium (LDC), University of Pennsylvania, Philadelphia, PA, USA*.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078.

Ronan Collobert and Jason Weston. 2007. Fast Semantic Extraction Using a Novel Neural Network Architecture. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 560–567, Prague, Czech Republic, June. Association for Computational Linguistics.

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

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12, November.

Kareem Darwish. 2014. Arabizi detection and conversion to arabic. In *In the Arabic Natural Language Processing Workshop, EMNLP*.

Chiang David, Mona Diab, Nizar Habash, Owen Rambow, and Safiullah Shareef. 2006. Parsing arabic dialects. In *Proceedings of EACL*, pages 369–376.

Marco Dinarelli and Loïc Grobol. 2019. Hybrid neural models for sequence modelling: The best of three worlds. *CoRR*. arXiv preprint 1909.07102.

Kevin Duh and Katrin Kirchhoff. 2005. Pos tagging of dialectal arabic: a minimally supervised approach. In *Proceedings of the acl workshop on computational approaches to semitic languages*, pages 55–62.

Rehab M Duwairi, Mosab Alfaqeh, Mohammad Wardat, and Areen Alrabadi. 2016. Sentiment analysis for arabizi text. In *2016 7th International Conference on Information and Communication Systems (ICICS)*, pages 127–132. IEEE.

Helmut Feldweg, Ralf Kibiger, and Christine Thielen. 1995. Zum sprachgebrauch in deutschen newsgruppen. *Osnabrücker Beiträge zur Sprachtheorie*, 50:143–154.

Chayma Fourati, Abir Messaoudi, and Hatem Haddad. 2020. Tunizi: a tunisian arabizi sentiment analysis dataset. *arXiv preprint arXiv:2004.14303*.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *Proc. of ICML*.

Elisa Gugliotta and Marco Dinarelli. 2020. Tarc: Incrementally and semi-automatically collecting a tunisian arabish corpus. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 6279–6286.

Nizar Habash, Fadhl Eryani, Salam Khalifa, Owen Rambow, Dana Abdulrahim, Alexander Erdmann, Reem Faraj, Wajdi Zaghouani, Houda Bouamor, Nasser Zalmout, et al. 2018. Unified guidelines and resources for arabic dialect orthography. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Nizar Habash. 2010. Introduction to arabic natural language processing. *Synthesis Lectures on Human Language Technologies*, 3(1):1–187.

Ahmed Hamdi, Alexis Nasr, Nizar Habash, and Núria Gala. 2015. Pos-tagging of tunisian dialect using standard arabic resources and tools. In *Proceedings of the Second Workshop on Arabic Natural Language Processing*, pages 59–68. Association for Computational Linguistics (ACL), July.

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

Go Inoue, Hiroyuki Shindo, and Yuji Matsumoto. 2017. Joint prediction of morphosyntactic categories for fine-grained arabic part-of-speech tagging exploiting tag dictionary information. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 421–431.

Mustafa Jarrar, Nizar Habash, Faeq Alrimawi, Diyam Akra, and Nasser Zalmout. 2017. Curras: an annotated corpus for the palestinian arabic dialect. *Language Resources and Evaluation*, 51(3):745–775.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.

William Labov. 1972. *Sociolinguistic Patterns*. Conduct and Communication. University of Pennsylvania Press, Incorporated.

Thomas Lavergne and FranÃ§ois Yvon. 2017. Learning the structure of variable-order crfs: a finite-state perspective. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 433–439. Association for Computational Linguistics.

Cédric Lopez, Sarah Zenasni, Eric Kergosien, Ioannis Partalas, Mathieu Roche, Maguelonne Teisseire, and Rachel Panckhurst. 2018. Extracting absolute spatial entities from sms: comparing a supervised and an unsupervised approach. *Language and the new (instant) media*.

Thang Luong, Quoc V. Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2016. Multi-task sequence to sequence learning. In *International Conference on Learning Representations*.

Mohamed Maamouri, Ann Bies, Tim Buckwalter, and Wigdan Mekki. 2004. The penn arabic treebank: Building a large-scale annotated arabic corpus. In *NEMLAR conference on Arabic language resources and tools*, volume 27, pages 466–467. Cairo.

Mohamed Maamouri, Tim Buckwalter, Dave Graff, and Hubert Jin. 2007. Fisher levantine arabic conversational telephone speech. *Linguistic Data Consortium, University of Pennsylvania, LDC Catalog No.: LDC2007S02*.

Mohamed Maamouri, Ann Bies, Sondos Krouna, Fatma Gaddeche, and Basma Bouziri. 2009. Penn arabic treebank guidelines. *Linguistic Data Consortium*.

Abir Masmoudi, Nizar Habash, Mariem Ellouze, Yannick Estève, and Lamia Hadrich Belguith. 2015. Arabic transliteration of romanized tunisian dialect text: A preliminary investigation. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 608–619. Springer.

Abir Masmoudi, Mariem Ellouze Khmekhem, Mourad Khrouf, and Lamia Hadrich Belguith. 2019. Transliteration of arabizi into arabic script for tunisian dialect. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 19(2):1–21.

Deepthi Mave, Suraj Maharjan, and Thamar Solorio. 2018. Language identification and analysis of code-switched social media text. In *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching*, pages 51–61.

Lesly Miculicich, Dhananjay Ram, Nikolaos Pappas, and James Henderson. 2018. Document-level neural machine translation with hierarchical attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2947–2954, Brussels, Belgium, October-November. Association for Computational Linguistics.

Etienne Morel and Simona Pekarek Doehler. 2013. Les 'textos' plurilingues: l'alternance codique comme ressource d'affiliation à une communauté globalisée. *Revue française de linguistique appliquée*, 18(2):29–43.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Rachel Panckhurst. 2017. *Entre linguistique et informatique. Des outils de traitement automatique du langage naturel écrit (TALNE) à l'analyse du discours numérique médié (DNM).* Ph.D. thesis, Université Paris-Est, Paris, France.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2012. Understanding the exploding gradient problem. *CoRR*, abs/1211.5063.

Arfath Pasha, Mohamed Al-Badrashiny, Mona Diab, Ahmed El Kholy, Ramy Eskander, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan Roth. 2014. MADAMIRA: A fast, comprehensive tool for morphological analysis and disambiguation of Arabic. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1094–1101, Reykjavik, Iceland, May. European Language Resources Association (ELRA).

Monica A Riordan and Roger J Kreuz. 2010. Emotion encoding and interpretation in computer-mediated communication: Reasons for use. *Computers in human behavior*, 26(6):1667–1673.

Oren Soffer. 2010. "silent orality": toward a conceptualization of the digital oral features in cmc and sms texts. *Communication Theory*, 20(4):387–404.

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

Natalie Sullivan. 2017. *Writing Arabizi: Orthographic Variation in Romanized Lebanese Arabic on Twitter*. Ph.D. thesis, The University of Texas at Austin, USA.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of NIPS*, Cambridge, MA, USA. MIT Press.

Benedikt Szmrecsanyi and Lieselotte Anderwald. 2018. Corpus-Based Approaches to Dialect Study. In *The Handbook of Dialectology*, pages 300–313. John Wiley & Sons, Ltd. Section: 17 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781118827628.ch17.

Toshiyuki Takezawa, Genichiro Kikui, Masahide Mizushima, and Eiichiro Sumita. 2007. Multilingual spoken language corpus development for communication research. In *International Journal of Computational Linguistics & Chinese Language Processing, Volume 12, Number 3, September 2007: Special Issue on Invited Papers from ISCSLP 2006*, pages 303–324.

Yasmin Tantawi and Mary Beth Rosson. 2019. The paralinguistic function of emojis in twitter communication. *Social Media Corpora for the Humanities (CMC-Corpora2019)*, page 68.

Sunisa Todla. 1999. Patterns of communicative behaviour in internet chatrooms. *Unpublished master's thesis, Chulalongkorn University*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*.

Fangzhao Wu and Yongfeng Huang. 2015. Collaborative multi-domain sentiment classification. In *2015 IEEE International Conference on Data Mining*, pages 459–468.

Felix Wu, Angela Fan, Alexei Baevski, Yann Dauphin, and Michael Auli. 2019. Pay less attention with lightweight and dynamic convolutions. In *International Conference on Learning Representations*.

Martin Wynne. 2005. *Developing linguistic corpora: A guide to good practice*. Oxbow Books Limited.

Simeon J. Yates. 1996. Oral and written linguistic aspects of computer conferencing. *Pragmatics and beyond New Series*, pages 29–46.

Jihen Younes, Hadhemi Achour, and Emna Souissi. 2015. Constructing linguistic resources for the tunisian dialect using textual user-generated contents on the social web. In *International Conference on Web Engineering*, pages 3–14. Springer.

Jihene Younes, Hadhemi Achour, Emna Souissi, and Ahmed Ferchichi. 2020. Romanized tunisian dialect transliteration using sequence labelling techniques. *Journal of King Saud University-Computer and Information Sciences*.

Nasser Zalmout and Nizar Habash. 2019. Joint diacritization, lemmatization, normalization, and fine-grained morphological tagging. *arXiv preprint arXiv:1910.02267*.

Inès Zribi, Mariem Ellouze Khemekhem, and Lamia Hadrich Belguith. 2013. Morphological analysis of tunisian dialect. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 992–996.

Inès Zribi, Mariem Ellouze, Lamia Hadrich Belguith, and Philippe Blache. 2017. Morphological disambiguation of tunisian dialect. *Journal of king Saud University-computer and information sciences*, 29(2):147–155.