LDK 2023

An Empirical Analysis of Task Relations in the Multi-Task Annotation of an Arabizi Corpus

Elisa Gugliotta and **Marco Dinarelli***
Univ. Grenoble Alpes, CNRS, Grenoble INP*, LIG, 38000 Grenoble, France;
* Institute of Engineering Univ. Grenoble Alpes; Groupe *getalp*
elisa.gugliotta@univ-grenoble-alpes.fr; marco.dinarelli@univ-grenoble-alpes.fr

Abstract

In this study, we deal with the design of computational-linguistic resources and strategies for the analysis of under-resourced languages. In particular, we present empirical analyses aiming at identifying the best path to semi-automatically annotate a dialectal Arabic corpus via a neural multi-task architecture. Such an architecture is used to automatically generate several levels of linguistic annotation which can be evaluated by comparison with the gold annotation. Changing the order in which annotations are produced can have an impact on the quantitative results. Through multiple sets of experiments we show how to get the best performances with this methodology.

1 Introduction

In this paper we present an empirical investigation of the relations between different levels of linguistic annotation of a dialectal Arabic corpus. In fact, linguistic annotations, such as Part-of-Speech (POS) tagging or lemmatisation, are an important prerequisite for many NLP applications and in particular, for those concerning under-resourced languages such as Arabic Dialects (ADs) (Elhadi and Alfared, 2022). The development of NLP resources and systems for under-resourced languages requires awareness of their functioning in order to study them from a computational perspective. This type of awareness derives from the analytical study of the language in question. However, while high-resourced languages present many detailed linguistic studies, often under-resourced languages usually lack comprehensive, in-depth and up-to-date descriptions of their morphological and syntactic systems. Moreover, they are often characterised by graphic variations and the lack of a standard orthography. In many cases, the spelling is not standardised and reflects geolinguistic variations (Bernhard et al., 2021).[1] This is also the case of the ADs, for which building resources such as linguistic annotated corpora, is a necessary stage to study and process them automatically. This is the reason why in the last couple of years there have been many projects focused on the creation of resources for the ADs.[2] A popular methodology to avoid the creation of AD corpora from scratch is the adaptation of resources, for example built for Modern Standard Arabic (MSA), in order to process ADs Harrat et al. (2018); El Mekki et al. (2021); Qwaider et al. (2019). However, MSA is used to perform language tasks completely different from those performed by using ADs. With this regards, Hary (1996) defines *multiglossia* as the linguistic situation in which different varieties coexist side-by-side in a language community, and where each variety is employed in different circumstances and has different functions. Therefore, in order to process ADs, the ideal solution should be to build dialect-centered resources from scratch, instead of adapting MSA resources, even though it involves a considerable effort. However, considering the enormous amount of work required to build resources from scratch, a possible strategy is adapting other existing AD tools to the AD under investigation, especially if the dialects belong to the same geographical areas (e.g. Tunisian and Algerian belong to the same area, namely the *Maghreb*). This is because ADs share much more with each other than with MSA.[3] In fact, a number of features and variations within ADs seem to transcend regional boundaries and effectively escape the most traditionally accredited typology, which classifies the ADs into six major dialectal areas, from East (*Mashreq*) to West (*Maghreb*). A possible explanation resides into the

* This article was prepared jointly by the two authors and is based on Gugliotta's post-doctoral research work supervised by Dinarelli. However, for the requirements of the Italian Academy, Gugliotta must be considered responsible for sections 1, 2.2, 3, 5.1 and 6, while sections 2.1, 4 and 5.2 must be attributed to Dinarelli.

[1] Common phenomena are variations in pronunciation, as well as morphological variations, where inflected or derived forms vary according to location, or lexical variations. Furthermore, the absence of standard spellings leads to interpersonal variation.

[2] See Ahmed et al. (2022) for a review on free Arabic corpora.

[3] For a study of the degree of similarity and dissimilarity between MSA and ADs, and among ADs, see Kwaik et al. (2018).

huge amount of migration, inter-dialectal contacts and many waves of diffusion which have brought specific linguistic features across the Arabic-speaking world (Benkato, 2019; Magidow, 2021; Benkato, 2020).

The creation of annotated corpora from scratch can be speed up by semi-automatic annotation using machine learning tools (Gugliotta and Dinarelli, 2020). In the case of multiple levels of annotation like in this work, a further benefit in using machine learning techniques can be obtained by exploiting Multi-Task (MT) learning, and in particular with neural models. MT neural learning approaches factorize information among learned tasks, improving results on all of them compared to individual tasks taken separately. Whether MT is performed in a parallel or cascaded fashion, it allows for sharing the representation of information of different tasks at intermediate layers (Caruana, 1997). MT has been proven to be particularly beneficial for ambiguous data, considering its ability to reduce sparsity, and helping to process complex patterns which involve multiple features. This is the case, for example, of POS-tagging (Rush et al., 2012; Søgaard and Goldberg, 2016; Alonso and Plank, 2016; Bingel and Søgaard, 2017; Hashimoto et al., 2016), which is particularly relevant to the morphological richness of Arabic, (as addressed by Inoue et al. (2017)) or dialectal Arabic (Zalmout and Habash, 2019).

For all these reasons and with the goal of basing our work particularly on AD, we found useful to exploit two resources recently created for the processing of Tunisian Arabic (Gugliotta and Dinarelli, 2022). The first resource is a MT neural architecture (see Section 2.1), built to help in annotating on multiple levels a Tunisian Arabizi Corpus. The second resource is the corpus itself (see Section 2.2). Concerning Arabizi, we must emphasize the spontaneous nature of this Roman orthography, which originated in digital environments where informal exchanges take place. Spontaneity plays a main role in the degree of encoding freedom left to native users, and this has an impact on the performance of MT systems. Other elements that play an influential part in MT learning systems include the design of the architecture itself and the order in which tasks are addressed. Beyond few exceptions, much of the existing work on MT learning systems focuses on learning one target task and one, or more, accurately selected auxiliary tasks (Changpinyo et al., 2018). There are various studies on multi-task learning, but it is not clear when this may be beneficial for all the tasks planned for the system, or when it may instead produce a phenomenon known as negative transfer, that also depends on the interrelations among the tasks (Ruder, 2017).[4] One of the keys to investigate this issue concerns the degree to which tasks are interrelated. A logical hypothesis is that morphological tasks may help syntactic tasks. With regard to the mentioned previous work on multi-task annotation, summarized in Gugliotta and Dinarelli (2022), the goal was to produce accurate annotations while facilitating manual checking work. Therefore, five levels of annotation were produced in a cascaded chain, via a MT learning system without delving, from a computational-linguistic point of view, into the degree of task interrelation. In this work, through exploiting these tools, we aim at finding possible task relations, and possibly improve previous results on each task by investigating such issue.

In order to explore this topic comprehensively, first of all, in Section 2, we will describe the architecture and the data on which we are relying for our study. Secondly, in Section 3, we will present the main related works. In Section 4, we will present the adopted methodology to address this issue. In Section 5, we will outline the experiments performed, drawing attention to some emerging trends. In the same section, we will discuss our results from a global point of view. Finally in Section 6 we will conclude the article.

2 MT Architecture and Data Structure

Like deep learning in general, multi-task learning is inspired by human learning. To learn new tasks, humans often transfer knowledge gained from prior related tasks. The possibility that certain cognitive structures may be prerequisites or have a positive or negative influence on the acquisition of new knowledge has been discussed by many researchers in the fields of didactics, pedagogy, cognitive linguistics, and psycholinguistics (Piaget, 2003; Vygotsky and Cole, 1978; Bransford and Johnson, 1972; Kole and Healy, 2007; Gick and Holyoak, 1980). However, the views of scholars are still too heterogeneous to explain the mechanisms and processes operating during human acts of comprehension and acquisition. Still it is well established that appropriate prior knowledge must be activated in order to be used effectively in the acquisition process. In a similar manner, Ruder (2017) motivates MT learning from the perspective of machine learning, viewing it as a form of inductive transfer. Indeed, the author explains that inductive

[4] See Section 3 for an outline of the existing work on MT learning systems and tasks interrelations.

transfer can help to improve a model by introducing an inductive bias, leading the model to prefer some assumptions over others. The inductive bias can be introduced by auxiliary tasks. Auxiliary tasks in MT learning can serve as conditions or suggestions for the main task. At the same time, related tasks can reinforce each other to form coherent predictions through shared representations. This strategy often leads to solutions that generalize better. However, according to Ruder (2017), our understanding of the degree of relationship or similarity between tasks is still limited, and we need to study them more in depth to better understand the generalization capabilities of MT learning by better fruiting their potential. Thus, one of the prerequisites of MT learning is the correlation between different tasks and data (Zhang et al., 2022).

2.1 The MT Architecture

The MT neural architecture employed in this work is an *encoder-decoder* system designed originally for the Tunisian Arabish Corpus (TArC) annotation. The MT system is able to instantiate as many decoders as the number of levels of linguistic annotations employed in the data, the different decoders operate in a cascade fashion, and it has been recently released (Gugliotta and Dinarelli, 2022). The MT system is designed to train LSTM or Transformer models. For our experiments we employed the LSTM model. As pointed out in (Gugliotta and Dinarelli, 2022), Transformers are in general preferred and very accurate for several NLP problems, especially when dealing with very large amount of data. However, they present limitations when modelling tasks with structured outputs (Weiss et al., 2018; Hahn, 2020). Since in our experiments outputs are always, at least partially structured, we employed mainly LSTM models. (Gugliotta and Dinarelli, 2022) shows indeed a significant performance gap between LSTM and Transformer models in experiments involving the TArC corpus, the same data we use in this work (please see the next section, for data description). Whatever the used model, the linguistic information that can be output by the MT system are: Code-Switching classification, normalization into CODA* (Habash et al., 2018), tokenization, POS-tagging and lemmatisation.

Concerning the classification of code-switching, it is provided at word level, in order to filter the Arabizi text from the foreign words, which are indeed classified as *foreign*. Table 1 presents the classification (**Class.** in the table header), the CODA* transliteration (**CODA***), the tokenization (**Token.**), the POS-tagging (**POS**) and the lemmatisation (**Lemma**) of the following Arabizi sentence of TArC.

(1) *Inchalah cycle ejjay wala eli ba3dou*,
/nšāḷḷa *cycle* əž-žāy walla əlli baʕd-u/,
'God willing next time, or the time after that'.

Arabizi	Class.	CODA*	Token.	POS	Lemma
Inchalah	Az.	ان شاء الله	ان شاء الله	INTERJ	ان شاء الله
cycle	Fr.	Fr.	Fr.	Fr.	Fr.
ejjay	Az.	الجاي	الـباي	DET+ADJ	جاي
wala	Az.	ولّا	ولّا	CONJ	ولّا
eli	Az.	اللي	اللي	REL_PRON	اللي
ba3dou	Az.	بعده	بعد ـه	ADV+ PRON_3MS	بعد

Table 1: Example of the annotation levels. "Az." means "Arabizi", "Fr." means "foreing".

Each of these annotation level is processed by a dedicated decoder. As for the Arabizi input, it is converted into context-aware hidden representations by the MT system's encoder. Each decoder is equipped with a number of attention mechanisms corresponding to the number of preceding modules (including the encoder). Hence, each decoder receives as input the hidden state of the encoder together with the hidden state of each previous decoder. Each decoder generates also its predicted output, which is used to learn the corresponding task by computing a loss function comparing the predicted output to the expected output. The entire architecture is learned end-to-end by calculating a global loss through the sum of each individual loss (Gugliotta and Dinarelli, 2022).

2.2 The Data

The data we used for the study presented in this paper are the TArC corpus (Gugliotta and Dinarelli, 2022) and the MADAR corpus (Bouamor et al., 2018). The first one contains 4,797 sentences produced by Tunisian users in digital contexts such as blogs, forums and social networks. These sentences are encoded in Arabizi, the Latin encoding employed for online written conversations. The MADAR corpus, on the other hand, is a parallel corpus of several Arabic dialects, including Tunisian (both from Tunis and Sfax cities). In our previous work, we exploited 2,000 sentences of the MADAR corpus, by proving their usefulness for the MT system learning (Gugliotta and Dinarelli, 2022). Also for experiments in this work we decided to use both corpora. In particular the MADAR data are concatenated to the TArC training data to create a single, bigger training set.

3 Related Work

Intuitively determining the degree of similarity between tasks is still a common practice especially in the design stages of MT architectures, when one does not yet have data on which to rely otherwise (Worsham and Kalita, 2020). In general, until a few years ago, methods for identifying task relationships focused on expert intuition. However, recent research increasingly takes into account the fact that neural networks do not need to operate on the same principles as human learning. More and more scholars, such as Alonso and Plank (2016), are arguing that the selection of MT learning tasks should be guided by the properties of the data, not by the intuition of what a human performer might consider easy. In fact, they conduct a number of studies showing that the best auxiliary tasks are neither too easy to predict nor too difficult to learn. In particular, for the mentioned study, they use a state-of-the-art architecture based on biLSTM models and evaluate its behavior on a motivated set of main and auxiliary tasks. The performance of the MT system is evaluated both by experimenting with different combinations of main and auxiliary tasks and by applying a frequency-based auxiliary task to a set of languages, processing tasks and evaluating its contribution. LSTM networks were also analyzed by Reimers and Gurevych (2017) for a wide variety of sequence tagging tasks, in order to find LSTM network architectures that can perform robustly on different tasks. Five classical NLP tasks were chosen as benchmark tasks: POS tagging, Chunking, Named Entity Recognition (NER), Entity Recognition and Event Detection. Guo et al. (2018) addressed multitask and curriculum learning to improve training of subsets of multiple tasks, starting with smaller and simpler tasks first. Zamir et al. (2018) computed an affinity matrix between tasks based on whether the solution of one task can be read easily enough by the representation trained for another task. Their approach, being fully computational and representation-based, avoids imposing prior (possibly incorrect) assumptions about the relationships between tasks. In addition, Standley et al. (2019), using the Taskonomy dataset (Zamir et al., 2018), found that, unlike affinities between transfer tasks, affinities between multiple tasks depend strongly on a number of factors such as dataset size and network capacity. A similar work to ours was presented by Bingel and Søgaard (2017), who conducted a study on ten traditional NLP tasks (including POS tagging), comparing the performance of MT and Single-Task (ST) learning, where hyperparameters of ST architectures are reused in the MT configuration. Changpinyo et al. (2018) conducted extensive empirical studies on eleven sequence labeling tasks. They obtained interesting pairwise relationships that reveal which tasks are beneficial or detrimental to each other. Such information correlated with MT learning outcomes using more than two tasks. They also studied the selection of only advantageous tasks for joint training, showing that this approach, in general, improves MT learning performance, and highlighting thus the need to identify tasks to be learned jointly. Similar experiments, but specific to the domain of question answering, were performed by Vu et al. (2020) who conducted an in-depth study of the relationships between various tasks (question answering and sequence tagging) and proposed a task-embedding framework to predict these relationships. Sun et al. (2020) sought to enable adaptive sharing by learning which levels are used by each task through model training. More recently, Aribandi et al. (2021) proposed a massive collection of various supervised NLP tasks in different domains and task families in order to study the effect of multi-task pre-training on the largest scale to date and analyze the transfer of co-training between common task families. The researchers addressed the issue of inter-language transfer from high-resource languages to low-resource languages. They presented a model capable of automatically selecting the language from which to transfer a given task, based on inter-lingual criteria. Fifty et al. (2021) proposed a procedure for selecting subtasks based on task gradients.

4 The Adopted Methodology

During the annotation process of the TArC (Gugliotta and Dinarelli, 2022), a specific order of linguistic annotation production has been set out. Starting from the Arabizi as input, this specific order was: classification (to filter the code-switching elements), transliteration into CODA*, tokenization, POS-tagging and lemmatisation. This order of annotation was chosen based on principles of both linguistic reasoning and empirical observation of MT system performances. Starting from the premise that providing too much information to an algorithm can slow it down and lead to inaccurate results, it is important to think carefully about what information is most relevant to a specific goal. The ultimate goals of Gugliotta and Dinarelli (2022) were **1.** to produce precise annotation levels and at the same time **2.** to ease the work of manually checking and correcting the annotations predicted by the architecture. Therefore, in Gugliotta and Dinarelli (2022),

the chosen order of tasks was oriented toward simplifying both the tasks involved in the semi-automatic annotation (the automatic classification and the manual correction). In fact, it was considered useful to find a good compromise between proceeding in hierarchical order, from the simplest to the most complex annotation (observing the performance of the MT system in annotation), while respecting the relationships between the various levels of annotation based on linguistic reasoning. Concerning the choice of processing easy tasks first, it is possible to define what is the easiest task among others for a model by observing its learning progress or the result precision in case of classification tasks (Guo et al., 2018). For example, as we noticed by observing experimental results in Gugliotta and Dinarelli (2022), the task of transliteration from Arabizi into CODA*, resulted to be the most difficult task for the architecture. In our opinion, this difficulty comes from the ambiguity of Arabizi, being a spontaneous orthographic system. On the other hand, it results more complicated to establish what task can be the most difficult for a human annotator, because this depends on his specific previous experiences, which are hard to evaluate and are in any case unlikely to match exactly the goal of the annotation at hand. For manual checking of data, for example, annotators will make use of their prior skills and the annotation guidelines, and they will apply this knowledge to the new task, gradually becoming faster and more effective. In fact, we can consider them as learners. As a result, if we apply the same logic as the one used in language acquisition theories, the ease of a task is closely related to the concept of support, in terms of knowledge, that is made available to perform the task.[5] This is to say that, for example during a manual correction phase, an annotator may find easier to correct various levels simultaneously, instead of correcting them one-by-one. Two possible reasons are (A.) the same error may have been transferred between different annotation levels, so it is easier to correct the various levels together. (B.) The presence of the other levels can help the annotator to better understand the error. The annotator will not only dispose of the text semantics, but also of the other levels of annotation (morpho-syntactic in the case of Gugliotta and Dinarelli (2022)). Therefore, generalising the problem, we might conclude that the "simple-to-complex" order can work as well for deep learning systems as for human learners (including annotators). However, as already mentioned in Section 3, we must consider that what is possibly an auxiliary task for an annotator does not help a MT learning system in the same way. The experiments in the following section are aimed at investigating this concern.

[5] Concerning human language acquisition knowledge there are several theories, like for example the one called Zone of Proximal Development (ZPD) (Vygotsky and Cole, 1978). The ZPD represents the interval between what a learner is able to do unsupported and what he can achieve with support. Support may come from someone else with wider knowledge or skills (namely the teacher).

4.1 Experimental Procedure

We organized different groups of experiments with the aim of identifying the best order of tasks to be performed by the architecture, and this in order to maximize the results on each of them. The first two groups of experiments are a mixture of ST (Single-Task) and MT (Multi-Task) strategies, organized into an iterative procedure. The procedure starts with using two annotation levels, one as input and the other as output task, where all possible combinations of two levels are tested to find the best order, results are shown in the tables 2 and 3. The order is thus chosen based on the best performing one. Performances on all tasks are measured with Accuracy (see Gugliotta and Dinarelli (2022)). Table 4 instead presents the grouping of particular intermediate experiments, in order to answer specific task relation questions. The iterative procedure continues using the annotation level detected as the easiest to predict, measured with empirical results, as the input to the system, and all the remaining annotation levels as output, both one at a time with ST experiments and with specific combinations of two or more annotation levels in MT experiments. This allows to select again the easiest task based on the empirical results. Results are given in the tables 5 and 6. We take care of using as much as possible Arabizi or CODA* as input to the system since these are the formats in which data may be naturally found, and needing to be transliterated, into CODA* for Arabizi and into Arabizi for CODA* (Gugliotta and Dinarelli, 2022), in addition to being annotated with the other levels of the TArC corpus (Gugliotta and Dinarelli, 2020) used also in this work for our analyses. Considering the spontaneous nature of Arabizi and the small amount of our data, having Arabizi text as input exposes to the risk of transferring errors obtained on the first task to the rest of the MT chain, hiding possibly the task-relation potential. For this reason, we performed two sets of experiments, one with Arabizi as input and one with Arabic script as input, the latter follows a conventional orthography (CODA*) and thus allows possibly to overcome the error transfer problem

implied by the use of Arabizi as input. The other experiments are based on MT learning. In fact, we want to compare the results obtained with the ST strategy with the same experiments performed in a MT setting. For these sets of experiments, we test different MT chains, that present different task orders, to observe which one is giving the best results, again testing both Arabizi (tables 7 and 10) and CODA* (tables 8 and 9) as input.

5 Results and Discussion

In this section we present the results of all our experiments. In Section 5.1 we present the preliminary experiments (mix of ST and MT strategies), while in the section 5.2 we present the results of the MT experiments. The experiments described in the Section 5.1, refer to a procedure centred on the observation of the best results of ST experiments, which then contribute to the definition of a precise task order in MT experiments. Therefore within this section, these MT experiments, which respect the order deduced from the ST results, will also be described. In order to provide a comprehensive description of the results and highlight the correlation between them, we will also globally discuss the results at the end of the paper (Section 6).

5.1 Preliminary Experiments

Table 2 shows our results on the test sets of TArC in the ST (Single-Task) experiment setting, using Arabizi and CODA* as input to the model.[6] We defined these experiments as the *Starting ST experiments*, considering them as the first stage to define a task order for the MT architecture. When the input was the Arabizi text we also performed the classification task (*(class.)* in the table header), in order to filter the code-switched tokens not to process. In the column *Arabizi input* we thus report also the classification accuracy for each experiment, in brackets. Experiments are performed using both Arabizi and CODA* as input since the system can be used in some cases to transliterate Arabizi data into CODA* encoding, like for the TArC corpus, in some cases for transliterating CODA* encoded data into Arabizi, like for the MADAR corpus (Gugliotta and Dinarelli, 2022), in addition to the other annotation levels when these are available to train the model for doing so.

The ST tasks performed for these experiments are the tokenization of the input, the Part-of-Speech tagging, the lemmatisation and the transliteration of Arabizi into CODA* (for the experiments having Arabizi as input), or of CODA* into Arabizi (in case of the experiments having CODA* as input). These tasks are reported in the table, in the column **Tasks**, with the respective entries: **Token.**, **POS**, **Lemma** and **Translit.**. Some results are in bold because they represent the best among the experiments reported within the table. As we can observe, both in the case of Arabizi and CODA* as input, the *easiest* task seems to be the tokenization, on which the system respectively achieved the accuracy of 80% and 95.4%. The former result is not surprising observing that, when using Arabizi as input, the transliteration task obtains one point less (79%) than the tokenization task (80%), these seem two very correlated annotation levels given the result on the tokenization task when using CODA* as input (95.4%). In fact, the tokenization implies the transliteration of the token, being both encoded in CODA* (as shown in Table 1). It is also interesting to observe the result on the classification task (93%) performed together with the tokenization, using Arabizi as input. Even if the difference is small, this is the best classification result. Thus, it seems that the classification benefits from the information of the tokenization task. It is also worth to highlight that both the tokenization and the lemmatisation performed from a CODA* input, obtain relatively high results, respectively 95.4% and 89.5%. While results on the POS (54.5%) and the transliteration into Arabizi (67.2%), using CODA* as input, are the lowest results, also compared to results obtained using Arabizi as input. Tokenisation and lemmatisation involve simpler processes than POS-tagging (identification of both the morphological class and the features of the token). In addition, we should consider that the CODA* conventional orthography is also employed to encode the tokenization and the lemmatisation levels. Indeed, these tasks result in *easy* operations for the model having as input the text in CODA*. This is not the case of the transliteration, where the system must convert the Arabic-encoded input into Latin-encoded information. In fact, it is surprising that the transliteration into CODA* is still obtaining a good result (79%) starting from an Arabizi input. This can be due to the fact that, as previously

Tasks	Arabizi input (class.)	CODA* input
Token.	**80.0% (93.0%)**	**95.4%**
POS	73.8% (92.5%)	54.5%
Lemma	75.5% (92.8%)	**89.5%**
Translit.	**79.0%** (92.8%)	67.2%

Table 2: Starting ST Experiments

[6] Please see Gugliotta et al. (2020); Gugliotta and Dinarelli (2022) for further details on the data and the architecture.

mentioned, the Arabizi encoding is a spontaneous, ambiguous script, while CODA* is a normalized encoding. Consequently, transliterating an ambiguous script into its normalization (i.e. many variations into one encoding) results to be an *easier* task in comparison to the opposite operation (CODA* into Arabizi, i.e., one encoding into one of the many encoding possibilities).

Once assessed that the tokenization task is the easiest using both Arabizi and CODA* as input, we continued the iterative procedure by using the detected easiest annotation level as input, and the other remaining annotation levels as output, both one at a time and all together in a MT learning setting. More precisely, we first performed ST experiments using the tokenization as input to the model, and alternatively POS and lemmas as output. These results are shown in the first two lines of Table 3, and they show that the easiest task between POS tagging and lemmatization, when using tokenization as input, is the lemmatization.

Input	Tasks	Accuracy
Token.	POS	86.2%
Token.	**Lemma**	**92.4%**
Token.	**Lemma - POS**	**92.8% - 87.6%**
Token.	POS - Lemma	87.3% - 92.6%

Table 3: Intermediate Experiments

By comparing the results of these two experiments, we can confirm our previous consideration about the fact that lemmatisation, in comparison to POS-tagging, is in general a simpler process to be performed starting from the token. The information that most helps the lemmatisation of a token is its morphological class. This information is contained in the POS, and more precisely in what we can define as the *main* part of the POS (namely only the morphological class, such as "verb", "noun", "adjective" etc., without its features, such as gender and number). The prediction of the *main* POS is a much easier task than the prediction of a POS with all the morphological features. In fact, the lemmatisation task obtains 92.4% of accuracy, 6.2 points more than the results on the POS tagging (86.2%). In the same table we also report two additional experiments that compare the combination of the two tasks (POS and lemmatisation) in the two possible orders, thus in a MT (Multi-Task) setting. We can observe that the combination achieving the best results is the first one (namely **Lemma - POS**), where the model obtained 92.8% and 87.6% of accuracy on the two tasks, respectively. While the margin of improvement is small with respect to the other possible order (POS - Lemma), this confirms that the lemmatization is the easiest task using tokenization as input. Moreover it is interesting to see that in the two MT experiments results are always better than those obtained with ST experiments. This means that the two tasks help each other, which is what we expect in a MT learning setting. Given these results, we considered useful to explore the question further by means of additional experiments, shown in Table 4.

Input	Tasks	Accuracy
CODA*	Lemma - POS	89.2% - 84.2%
CODA*	**POS - Lemma**	**85.9% - 90.5%**
CODA*	Token. - POS	95.3% - 85.2%
CODA*	POS - Token.	85.6% - 95.2%

Table 4: Additional Experiments for Tasks Relations

These experiments present the grouping of particular intermediate annotation levels, using CODA* as input to the system. The aim of the experiments was to discover what task, between lemmatisation and tokenization, helped more the POS task, and in which order. For this reason we needed the tokenization not to be the input for the model, and among Arabizi and CODA* we preferred to have the input in CODA* in order to avoid introducing a bias in these experiments due to the errors depending on the Arabizi ambiguity. If we had to guess which task helps POS prediction more, we would have chosen tokenization rather than lemmatisation. The former is in fact a morphological task, as much as POS, while the latter is primarily a lexical (but also morphological) task. However, by observing the results, we can confirm what already observed in Table 3, namely that it is the lemmatisation the task helping more the POS tagging. In fact, the experiment showing the best results on POS is the second one, where the POS is followed by the lemmatisation. This result on the specific order (POS-Lemma) seems to be inconsistent with what has just been stated by commenting on Table 3, where slightly better results were obtained by keeping the Lemma-POS order. However, what makes the difference between the experiments in the tables 3 and 4 is the input. That is, when the input is the tokenized text, the Lemma-POS and POS-Lemma order obtain similar results (Table 3), whereas when the input is in CODA* (Table 4) there is a considerable difference in the two possible orders between POS and Lemma (POS improves of 1.7 accuracy points, Lemma improves of 1.3 points with the POS-Lemma order). Instead, we have non-significant differences by inverting the order between *Token.* and POS. Thus, it seems that the system has more difficulties in extracting the

	Accuracies on tasks			
Exp. ID	**Token.**	**Lemma**	**POS**	**Arabizi**
I	95.4%	-	-	-
II	95.3%	89.8%	-	-
III	**96%**	**90.7%**	**86.2%**	-
IV	94.4%	88.9%	84.5%	67.8%

Table 5: Chain based on ST experiments - CODA* input

	Accuracies on tasks				
Exp. ID	**Class.**	**Token.**	**Lemma**	**POS**	**CODA***
I	86.2%	-	-	-	-
II	93%	**80%**	-	-	-
III	95%	**80%**	**78.2%**	-	-
IV	94.1%	78.9%	77.5%	77.8%	-
V	**94.2%**	78.9%	77.3%	**78.6%**	**79.5%**

Table 6: Chain based on ST experiments - Arabizi input

lemma from the CODA*, without the intermediate step of POS tagging, which instead obtains better results (85.9%) directly on the CODA*, than on the lemma (84.2%), also helping to improve the results on the lemmatisation, which rises by 1.3 points (90.5% vs 89.2%), if placed after the POS level. This is also evident if we compare these results with those obtained in the ST experiment (CODA* - Lemmatization: 89.5%) in Table 2. The results on lemmatisation improve (by 1 point) when it follows the POS task (90.5%), thus, the two tasks (POS and lemmatisation) help each other. Once these considerations have been made, we can present the results in Table 5 and Table 6, that present the experiments aiming at identifying the final MT learning chain based on ST experiments, having CODA* (Table 5), or Arabizi (Table 6) as input. The progressive Roman numerals in the '**Exp. ID**' columns of these tables indicate the sequential order in which the experiments were performed. These numerals will also be used to refer to the experiments while discussing the results. Concerning these experiments, both in the case of an input in Arabic characters (CODA*) and in the case of an input in spontaneous Latin orthography (Arabizi), it emerges a tendency for improved results due to the presence of auxiliary tasks. With regards to Table 5, we can observe that thanks to the presence of the tokenization task, the lemmatisation improves of 0.3 points at the experiment II (second line of Table 5), in comparison with the lemmatisation experiment as a ST in Table 2. Observing the experiment III (*Exp.* from now on for short) reported in Table 5, we can notice that thanks to the presence of the POS task, the tokenization task improves of 0.7 points with respect to the ST experiment on tokenization, reported in Table 2.

Also the lemmatisation task obtains better results, improving by 0.9 points, thanks to the presence of the POS task, at the Exp. III, in comparison with the Exp. II in Table 5. Finally the transliteration task from CODA* into Arabizi improves by 0.6 points, thanks to the previous tasks (at the Exp. IV in Table 5, in comparison to the transliteration as an ST experiment in Table 2). However, by adding the transliteration to the chain of tasks, the model is subject to much more difficulty, as can be noticed at the Exp. IV of Table 5, where all the previous tasks undergo the negative transfer effect, due to the presence of the transliteration into Arabizi.[7] From Table 6 we can draw very similar observations. From the Exp. II, we can observe an improvement of 6.8 points of the classification task, in comparison with the Exp. I, thanks to the tokenization task. On the next step (Exp. III), classification continues to improve (by 2 points) thanks to the lemmatisation task, which also improves by 2.7 points (thanks to the tokenization) in comparison with the ST experiment on lemmatisation in Table 2. Finally, at the Exp. V, we can observe how, thanks to the normalization of Arabizi into CODA*, POS-tagging improves of almost one point (0.8), in comparison with the previous step (Exp. IV) in Table 6. Also the transliteration task obtains better results, 0.5 points in comparison with the ST transliteration reported in Table 2, thanks to the previous tasks. By observing Table 6, *the most difficult task* for the model seems to be the POS tagging. In fact, at the Exp. IV, while the POS task improves by an impressive 4.8 points (in comparison with the ST experiment in Table 2), all the previous tasks lose about one point, compared with the results of the previous step (Exp. III).

5.2 Multi-Task Experiments

In our multi-task system, as previously stated, variables come into play, such as the factorization of the information shared among the decoders, the presence of attention mechanisms, etc. For this reason, we decided to compare the results obtained from ST experiments with those of the MT experiments. Therefore, in the following tables we can observe different combinations of tasks performed sequentially by the MT architecture. The goal is to check whether or not the ST task-chain matches with the MT task-chain that gives better results than other combinations or than the combinations that would seem logical from a linguistic point of view (e.g.: Arabizi - Classification - CODA* - Lemmatization - Tokenization - POS). Each

[7] This phenomenon has been mentioned in the section 1.

line of the following tables represents an experiment with all the tasks in a specific order. The order of a task is specified in brackets as a footnote of the corresponding accuracy result. When such note is not present, the order of the task is the one corresponding to the column of the table. For instance in the Exp. I in Table 7, the task order is *Class.* - *CODA** - *Token.* - *POS* - *Lemma*, where the order of Class. and CODA* is the one given by the corresponding column in the table since their accuracy has no footnote; while for *Token.*, *POS* and *Lemma* the order is given by the index in footnote to their accuracy. This notation allows to give several task orders in the same table keeping the same table headers. We also keep the same experiment identifier naming with roman cardinals as in the previous tables, e.g. Exp. I mentioned above.

Table 7 presents the MT experiments with the Arabizi text as input. For the experiments reported in this table, the first tasks are always the classification and the transliteration into CODA*. Concerning the last two line of the table (lines VII and VIII), they summarize the results of two experiments, where the model receives the Arabizi input and processes the tasks of lemmatisation and transliteration into CODA* as a second and third task, respectively.

	Accuracies on tasks				
Exp. ID	**Class.**	**CODA***	**Lemma**	**Token.**	**POS**
I	97.3	82.6	82.3(5)	82.3(3)	71.4(4)
II	**99**	**84.2**	82.8(4)	**83.5**(3)	**83.1**(5)
III	92.9	78.5	54.2(4)	75.9(5)	78(3)
IV	94.3	78.3	76.4(5)	77.9(4)	78.1(3)
V	97.9	**84.3**	**83.6**(3)	82.3(4)	82.3(5)
VI	98.8	83.5	82.4(3)	82.3(5)	82.3(4)
VII	98.5	83.7(3)	83(2)	83(4)	82.6(5)
VIII	93.2	77.8(3)	78.6(2)	76(5)	78.2(4)

Table 7: Chain based on MT experiments - Arabizi input

At the end of the section 5.1, by discussing the preliminary experiments, we stated that POS-tagging is the most difficult task, together with the transliteration into Arabizi. In particular, we have deducted this by looking at Table 6. In fact, we remind that for these experiments we imposed a task order based on ST experiments described in the section 4.1. We also recall that, in Table 6 (experiments concerning the MT-chain based on ST experiments) the highest result obtained on POS tagging was 78.6%.

Concerning the Multi-Task (MT) experiments and looking at Table 7, we can see that the highest result on POS is 83.1%. We can also note that on all tasks, except for lemmatisation, better results are achieved with the Exp. II, where POS is the last task processed by the MT architecture. Thus, it seems that POS prediction is benefiting of all the previous task information. The POS results in the Exp. II (83.1%) are improved of 4.5 points in comparison with the best result of Table 6 (78.6%). At the Exp. II, it is also interesting to observe how the lemmatisation task, processed between tokenization and POS, contributes to the improvement of both tokenization and POS, though it loses almost one point (0.8) compared to its highest result, obtained when lemmatisation is in the third position (see the Exp. V). In fact, at the Exp. V in Table 7, we can see that lemmatisation improves by 0.8 points if it follows the transliteration task and if it is followed by the tokenization task. The difficulty introduced by the POS task is evident from the tables 6, 7 and 3. In the latter one we also observed the encouraging results obtained on the lemmatisation task, using tokenization as input.

We also performed the experiments reported in Table 8, in order to identify the best task sequence for predicting Arabizi strings from CODA* strings. Considering that the input for these experiments is already filtered by the *foreign* tokens, we did not perform the classification task. Except for the transliteration into Arabizi, which is always the last task, the order of the tasks for each experiment are shown again through footnotes with a number in brackets.

	Accuracies on tasks			
Exp. ID	**Lemma**	**Token.**	**POS**	**Arabizi**
I	88.9(3)	94.8(1)	84.1(2)	68
II	88.9(2)	94.4(1)	84.5(3)	67.8
III	**89.7**(2)	**95.1**(3)	**85.1**(1)	**68.5**
IV	89.4(3)	94.7(2)	84.6(1)	**68.4**
V	**89.7**(1)	**95**(2)	84.7(3)	68.2
VI	89.1(1)	**95.2**(3)	**85**(2)	**68.4**

Table 8: Chain based on MT experiments - CODA* input

Even in Table 8 we can observe that MT experiments produced better results if compared to those of the task sequence established with the ST logic in Table 5. In fact, we defined the transliteration into Arabizi as the most complex task starting from an input in CODA*. In Table 5 the result obtained on transliteration was 67.8%, while in Table 8 we can see how in several experiments we obtained better results, and in general on all tasks. The chain established through the sequential logic of ST experiments, shown again in Table 8 as Exp. II, actually appears to be the worst combination for both tokenization and transliteration. We note, on the other hand, that the best over all tasks is the one that, in the Exp. III, sees POS in the first position of the task chain. Again, like

in Table 7, POS is separated from the rest of the tasks by the intermediate presence of the lemmatisation task, and followed by tokenization. It is very interesting to observe that in Exp. III POS gets as much as one point more than in the Exp. I of the same table, where it was the second task, after the tokenization task. We remind that according to the linguistic logic, the tokenization being a morphological task, it should support the morpho-syntactic tasks.

Finally, we performed experiments with different task combinations, considering the possibility that annotations, such as lemmas or POS-tags, are introducing negative a bias for the task of CODA* transliteration into Arabizi encoding, and that the classification can instead help in it. These are reported in Table 9. Concerning the experiments reported in the last two lines of the table (lines VII and VIII), these treated the lemmatisation as a second task, after the classification (which is always the first task) and before the task of transliteration into Arabizi. In fact, the latter is always the second task performed during the previous experiments reported in the same table (experiments 1-6).

	Accuracies on tasks				
Exp. ID	**Class.**	**Lemma**	**Token.**	**POS**	**Arabizi**
I	97.2	88.8(5)	94.5(3)	83.6(4)	**68.8**(2)
II	98.1	89.3(4)	95.3(3)	83.4(5)	68.3(2)
III	98.1	89.1(4)	95.2(5)	83.4(3)	68.5(2)
IV	97.4	88.6(5)	94.7(4)	83.3(3)	68.4(2)
V	97.8	88.9(3)	95.2(4)	84.3(5)	68.7(2)
VI	97.5	89.2(3)	94.4(5)	83.4(4)	68.3(2)
VII	97.5	89.3(2)	95(4)	83.6(5)	68.7(3)
VIII	**98.3**	89.2(2)	**95.4**(5)	**84.8**(4)	68.6(3)

Table 9: Other MT experiments to predict Arabizi

The goal of experiments reported in Table 10, instead, is to predict the CODA* transliteration from the Arabizi input. Thus, the transliteration into CODA* is always the last task, while the classification is always the first task.

	Accuracies on tasks				
Exp. ID	**Class.**	**Lemma**	**Token.**	**POS**	**CODA***
I	94.1	76.3(4)	77.9(2)	77.9(3)	78.1
II	94.2	77.3(3)	**78.9**(2)	78.6(4)	**79.5**
III	94	77.2(3)	78.2(4)	78.5(2)	78.2
IV	93.8	76.3(4)	78.1(3)	78.1(2)	78
V	94	77.2(2)	78.4(3)	78.5(4)	78.5
VI	94.2	77.3(2)	78.7(4)	78.8(3)	78.7

Table 10: Other MT experiments to predict CODA*

In these last two tables, 9 and 10, we have reported, for the sake of completeness, experiments with additional combinations of tasks. Both seem to confirm the concept with which we would like to conclude our analysis. Namely, specific task ordering in a MT learning setting, in the case of a robust model provided with attention mechanisms, matters up to a certain point. In fact, looking at the last two tables, where we aimed at improving transliteration into Arabizi (Table 9) and CODA* (Table 10), we can notice first that the tasks exhibit roughly always the same accuracy values in all experiments. As a second observation, two different strategies are adopted. In Table 9 the transliteration task in Arabizi is always in the second position (except for experiments VII and VIII), while in Table 10 transliteration in CODA* is always the last task. By comparing the results of the strategy in Table 9 with those obtained on the Arabizi transliteration task in Table 8 (where Arabizi is always the last task), we can say that the strategy of tackling Arabizi as the second task yields better results, although the difference is small. We can draw the same conclusion by looking at the results on the transliteration task into CODA*, comparing the results in Table 7 to those in Table 10. In the former, transliteration is always addressed as the second task (except in the experiments VII and VIII), and doing so yields better results than those reported in Table 10, where the transliteration task is always the last one.

6 Conclusions

In this work, we presented empirical analyses in order to pinpoint the best approach for semi-automatic annotation of a dialectal Arabic corpus through a multi-task neural architecture. The experiments performed highlight a number of factors that may play a role in the outcome of good data annotation. Among the ones discussed are the interrelations between the tasks processed by the architecture, the difficulty the architecture faces in performing the tasks and the impact that determining specific orders of data annotation may have on the results, especially if to infer the relationship between tasks, we rely *only* on linguistic intuitions. By observing the experiments performed by this study, it clearly emerges the existence of relations between tasks, and these are especially evident when observing ST experiments. In fact, it turned out that morphological information does not necessarily support morphological tasks (Table 4), whereas it supports, for example, lemmatisation. At the same time, lemmatisation appears to play a key role in supporting the POS task, which difficulty is evident from the tables 6, 7 and 3. In the latter one we also observed

the encouraging results obtained on the lemmatisation task, using tokenization as input. The optimal choice therefore is to isolate the POS task, leaving it as the last task to be processed and preceding it by all simple tasks such as tokenization or lemmatisation. The latter is probably more effective, as intermediate task between tokenization and POS, in that it consists in fewer operations to be performed by the model, which is then able to generalize better on lemmatisation, especially once the tokenization is performed as a previous task (see Table 3). In other words, the lemmatisation task, positioned between tokenization and POS, can provide a cushioning effect to the negative transfer introduced by the POS task (see for example the POS negative transfer effects on the tokenization at the Ex. V in Table 7). We also remind that, in section 5.1, by observing Table 4, we noted that: (**1.**) The best results on the POS, having the input in CODA, are obtained at the experiment where the POS is side-by-side with the lemmatisation instead of the tokenization. (**2.**) The accuracy on lemmatisation improves (by 1 point) in comparison with the ST accuracy (Table 2). This seems to mean that the reason why the lemmatisation level succeeds in "absorbing" the negative transfer of POS-tagging on the rest of the MT system, lies in two reasons. The **first** is that lemmatisation, basically, is an easy task (especially if based on CODA* transliteration, as shown in the tables 2 and 5), and the **second** is that the operations to perform POS-tagging are essentially a prerequisite to those implemented to solve the lemmatisation task. In fact, although POS-tagging is a complex task, it does not affect the lemmatisation results (as it does instead with the other tasks), actually POS improves the lemmatisation by disambiguating the string. In short, the two tasks are strongly related. However, imposing specific orders on tasks, according to such relations in ST learning logic has been shown to be an uncertain strategy in comparison to the MT strategy. Regarding the latter, we believe that what really has an influence on the results in terms of improvement of individual tasks is not so much the relation between tasks, but the inherent difficulty of tasks. In fact, there seems to be a tendency for general improvement in results on the various tasks if the tasks that require greater architectural capacity are tackled at the initial positions in the chain of tasks.

References

Arfan Ahmed et al. 2022. Free and accessible Arabic corpora: A scoping review. *Computer Methods and Programs in Biomedicine Update*, page 100049.

Héctor Martínez Alonso and Barbara Plank. 2016. When is multitask learning effective? Semantic sequence prediction under varying data conditions. *arXiv preprint arXiv:1612.02251*.

Vamsi Aribandi, Yi Tay, Tal Schuster, Jinfeng Rao, Huaixiu Steven Zheng, Sanket Vaibhav Mehta, Honglei Zhuang, Vinh Q Tran, Dara Bahri, Jianmo Ni, et al. 2021. Ext5: Towards extreme multi-task scaling for transfer learning. *arXiv preprint arXiv:2111.10952*.

Adam Benkato. 2019. From medieval tribes to modern dialects: On the afterlives of colonial knowledge in Arabic dialectology. *Philological Encounters*, 4(1-2):2–25.

Adam Benkato. 2020. Maghrebi Arabic. *Arabic and contact-induced change*, 1:197.

Delphine Bernhard et al. 2021. Collecting and annotating corpora for three under-resourced languages of france: Methodological issues. *Language Documentation & Conservation*, 15:316–357.

Joachim Bingel and Anders Søgaard. 2017. Identifying beneficial task relations for multi-task learning in deep neural networks. *arXiv preprint arXiv:1702.08303*.

Houda Bouamor et al. 2018. The madar Arabic dialect corpus and lexicon. In *Proceedings of the Eleventh International Conference on LREC*.

John D Bransford and Marcia K Johnson. 1972. Contextual prerequisites for understanding: Some investigations of comprehension and recall. *Journal of verbal learning and verbal behavior*, 11(6):717–726.

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

Soravit Changpinyo, Hexiang Hu, and Fei Sha. 2018. Multi-task learning for sequence tagging: An empirical study. *arXiv preprint arXiv:1808.04151*.

Abdellah El Mekki et al. 2021. Domain adaptation for Arabic cross-domain and cross-dialect sentiment analysis from contextualized word embedding. In *Proceedings of the 2021 Conference of the North American Chapter of the ACL: Human Language Technologies*, pages 2824–2837.

Mohamed Taybe Elhadi and Ramadan Alsayed Alfared. 2022. Adopting Arabic taggers to annotate a libyan dialect text with a pre-tagging processing and term substitutions. *ISTJ*.

Chris Fifty, Ehsan Amid, Zhe Zhao, Tianhe Yu, Rohan Anil, and Chelsea Finn. 2021. Efficiently identifying task groupings for multi-task learning. *Advances in Neural Information Processing Systems*, 34:27503–27516.

Mary L Gick and Keith J Holyoak. 1980. Analogical problem solving. *Cognitive psychology*, 12(3):306–355.

Elisa Gugliotta and Marco Dinarelli. 2020. TArC: Incrementally and Semi-Automatically Collecting a Tunisian Arabish Corpus. In *Proceedings of the Twelfth LREC*, pages 6279–6286.

Elisa Gugliotta and Marco Dinarelli. 2022. TArC: Tunisian Arabish Corpus first complete release. In *Proceedings of the Thirteenth International Conference on LREC*.

Elisa Gugliotta, Marco Dinarelli, and Olivier Kraif. 2020. Multi-task sequence prediction for Tunisian Arabizi multi-level annotation. *arXiv preprint arXiv:2011.05152*.

Michelle Guo, Albert Haque, De-An Huang, Serena Yeung, and Li Fei-Fei. 2018. Dynamic task prioritization for multitask learning. In *Proceedings of the European conference on computer vision (ECCV)*, pages 270–287.

Nizar Habash et al. 2018. Unified guidelines and resources for Arabic dialect orthography. In *Proceedings of the Eleventh International Conference on LREC*.

Michael Hahn. 2020. Theoretical limitations of self-attention in neural sequence models. *Transactions of the ACL*, 8:156–171.

Salima Harrat, Karima Meftouh, and Kamel Smaïli. 2018. Maghrebi Arabic dialect processing: an overview. *Journal of International Science and General Applications*, 1.

Benjamin Hary. 1996. The Importance of the Language Continuum in Arabic Multiglossia. *Understanding Arabic: essays in contemporary Arabic linguistics in honor of El-Said Badawi*, pages 69–90.

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

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.

James A Kole and Alice F Healy. 2007. Using prior knowledge to minimize interference when learning large amounts of information. *Memory & Cognition*, 35(1):124–137.

Kathrein Abu Kwaik, Motaz Saad, Stergios Chatzikyriakidis, and Simon Dobnika. 2018. A lexical distance study of Arabic dialects. *Procedia computer science*, 142:2–13.

Alexander Magidow. 2021. The old and the new: Considerations in Arabic historical dialectology. *Languages*, 6(4):163.

Jean Piaget. 2003. *The psychology of intelligence*. Routledge.

Chatrine Qwaider, Stergios Chatzikyriakidis, and Simon Dobnik. 2019. Can modern standard Arabic approaches be used for Arabic dialects? sentiment analysis as a case study. In *Proceedings of the 3rd Workshop on Arabic Corpus Linguistics*, pages 40–50.

Nils Reimers and Iryna Gurevych. 2017. Reporting score distributions makes a difference: Performance study of lstm-networks for sequence tagging. *arXiv preprint arXiv:1707.09861*.

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

Alexander M Rush, Roi Reichart, Michael Collins, and Amir Globerson. 2012. Improved parsing and pos tagging using inter-sentence consistency constraints. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 1434–1444.

Anders Søgaard and Yoav Goldberg. 2016. Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the 54th Annual Meeting of the ACL (Volume 2: Short Papers)*, pages 231–235.

T Standley, AR Zamir, D Chen, L Guibas, J Malik, and S Savarese. 2019. Which tasks should be learned together in multi-task learning? arxiv e-prints. *arXiv preprint arXiv:1905.07553*.

Ximeng Sun, Rameswar Panda, Rogerio Feris, and Kate Saenko. 2020. Adashare: Learning what to share for efficient deep multi-task learning. *Advances in Neural Information Processing Systems*, 33:8728–8740.

Tu Vu et al. 2020. Exploring and predicting transferability across nlp tasks. *arXiv preprint arXiv:2005.00770*.

Lev Semenovich Vygotsky and Michael Cole. 1978. *Mind in society: Development of higher psychological processes*. Harvard university press.

Gail Weiss, Yoav Goldberg, and Eran Yahav. 2018. On the practical computational power of finite precision RNNs for language recognition. In *Proceedings of the 56th Annual Meeting of the ACL (Vol 2: Short Papers)*, pages 740–745, Melbourne, Australia. ACL.

Joseph Worsham and Jugal Kalita. 2020. Multi-task learning for natural language processing in the 2020s: where are we going? *Pattern Recognition Letters*, 136:120–126.

Nasser Zalmout and Nizar Habash. 2019. Adversarial multitask learning for joint multi-feature and multi-dialect morphological modeling. *arXiv preprint arXiv:1910.12702*.

Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. 2018. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3712–3722.

Zhihan Zhang, Wenhao Yu, Mengxia Yu, Zhichun Guo, and Meng Jiang. 2022. A survey of multi-task learning in natural language processing: Regarding task relatedness and training methods. *arXiv preprint arXiv:2204.03508*.