ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

School of Information Sciences and Technology
Department of Informatics
Athens, Greece

Bachelor Thesis
in
Computer Science

A Transformer-based Natural Language Processing Toolkit for Greek – Part of Speech Tagging and Dependency Parsing

Chrysoula Dikonimaki

Supervisors: Ion Androutsopoulos
John Koutsikakis

July 2021

Chrysoula Dikonimaki

A Transformer-based Natural Language Processing Toolkit for Greek – Part of Speech Tagging and Dependency Parsing

July 2021

Supervisors: Ion Androutsopoulos, John Koutsikakis

Athens University of Economics and Business
School of Information Sciences and Technology
Department of Informatics
Natural Language Processing Group
Athens, Greece

Abstract

Named Entity Recognition (NER), Part-of-Speech (POS) tagging and Dependency Parsing are tasks that are central to Natural Language Processing (NLP) because they provide important information that can be used for other higher level NLP tasks. We develop a series of Deep Learning models that can perform these tasks in Greek. We experiment with GreekBERT, a pre-trained Transformer model for Greek, and XLM-R, a multi-lingual pre-trained Transformer model, providing additional training to (fine-tuning) both models in Greek for the tasks we consider. We find that using GreekBERT leads to better results in Dependency Parsing. In NER and POS tagging, GreekBERT and XLM-R lead to similar results. This work was combined with the work of the BSc thesis of Nikolaos Smyrnioudis [Smy21]. The two theses jointly developed a publicly available Transformer-based toolkit for Greek NLP. In this thesis, the emphasis was mostly in Part-of-Speech tagging as well as Dependency Parsing.

Περίληψη

Η αναγνώριση ονομάτων οντοτήτων (Named Entity Recognition), η κατηγοριοποίηση λέξεων σε μέρη του λόγου (Part-of-Speech Tagging, POS tagging) και η συντακτική ανάλυση εξαρτήσεων (Dependency Parsing) είναι προβλήματα πολύ κεντρικά στην επεξεργασία φυσικής γλώσσας (ΕΦΓ, Natural Language Processing, NLP) γιατί η επίλυση τους μας προσφέρει πολύτιμη πληροφορία που μπορεί να χρησιμοποιηθεί σε άλλα προβλήματα ΕΦΓ. Σε αυτή την εργασία αναπτύσσουμε μια σειρά μοντέλων βασισμένων στην Βαθιά Μάθηση (Deep Learning) για την επίλυση των συγκεκριμένων προβλημάτων στην ελληνική γλώσσα. Πειραματιζόμαστε με το GreekBERT, ένα προ-εκπαιδευμένο μοντέλο Transformer για την ελληνική γλώσσα, καθώς και με το XLM-R, ένα πολύγλωσσο προεκπαιδευμένο μοντέλο Transformers, παρέχοντας συμπληρωματική εκπαίδευση (fine-tuning) και στα δύο μοντέλα για τα προβλήματα που εξετάζουμε. Η χρήση του GreekBERT οδηγεί σε καλύτερα αποτελέσματα στη συντακτική ανάλυση εξαρτήσεων. Στην αναγνώριση ονομάτων οντοτήτων (NER) και στην κατηγοριοποίηση σε μέρη του λόγου (POS tagging), το GreekBERT και το XLM-R οδηγούν σε παρόμοια αποτελέσματα. Αυτή η εργασία συνδυάστηκε με την πτυχιακή εργασία του Νικόλαου Σμυρνιούδη. Οι δύο εργασίες ανέπτυξαν από κοινού μια δημόσια διαθέσιμη εργαλειοθήκη ΕΦΓ για την ελληνική γλώσσα, βασισμένη σε Transformers. Στην παρούσα εργασία, η έμφαση δίδεται στην κατηγοριοποίηση λέξεων σε μέρη του λόγου και στην συντακτική ανάλυση εξαρτήσεων.

Acknowledgements

I would like to thank everyone that has supported me with materials, advice and resources during the work of this thesis. I would like to especially thank Professor Ion Androutsopoulos for providing me with important ideas and learning resources for my thesis. Many thanks also go to John Koutsikakis for assisting with the technicalities of implementing Deep Learning models with PyTorch and continuous advice and supervision throughout my work. Many thanks also to Katerina Korre, for proof-reading and writing advice, to Manolis Kyriakakis for providing code and guidance with the implementation of the Dependency Parsing model and to Mary Georgiou for assisting with the development of the toolkit.

Contents

1 Introduction

Named Entity Recognition (NER), Part-of-Speech (POS) tagging and Dependency Parsing are tasks that are central to Natural Language Processing (NLP) because they provide important information that can be used for other NLP tasks. We develop a series of Deep Learning models that can perform these tasks in Greek. We experiment with GreekBERT [Kou+20], a pre-trained Transformer [Vas+17] model for Greek, and XLM-R [Con+19a], a multi-lingual pre-trained Transformer model, providing additional training to (fine-tuning) both models in Greek for the tasks we consider. This work was combined with the work of the BSc thesis of Nikolaos Smyrnioudis [Smy21]. The two theses jointly developed a publicly available Transformer-based toolkit for Greek NLP. In this thesis, the emphasis was mostly in Part-of-Speech tagging and Dependency Parsing."

1.1 POS tagging

Part of speech tagging (POS tagging or grammatical tagging), as Jurafsky et al. (2017) [JM17] define it, is the process of taking a sequence of words and assigning each word a part of speech like NOUN or VERB.

Some words are ambiguous and it is impossible to assign a specific POS tag to them if we don't fully understand the context. For example, the word book can be either a noun or a verb. It depends on the sentence it appeared. Knowing the POS tags of words can help in several other important tasks, such as NER [Far+08], dependency parsing, information retrieval [CM98] and text-to-speech synthesis [SB11].

Sentence	Pos tags
Janet	NOUN
will	AUX
back	AUX
the	DET
bill	NOUN

Tab. 1.1: Example of POS tagging from [JM17].

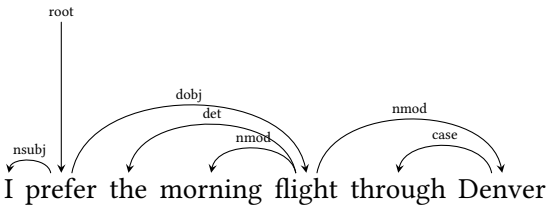


Fig. 1.1: Dependency tree from [JM17].

Relations	Description
NSUBJ	Nominal subject
CCOMP	Clausal complement
NMOD	Nominal modifier
DET	Determiner
CASE	Prepositions, postpositions and other case markers
CONJ	Conjunct

Tab. 1.2: Selected dependency relations from [De +14]

1.2 Dependency Parsing

In order to understand what Dependency Parsing is, we have to know what dependency trees are. As Jurafsky et al. (2017) [JM17] say, a dependency tree describes the syntactic structure of a sentence in terms of the words/lemmas in a sentence and an associated set of directed binary grammatical relations that hold among the words. Relations among the words are illustrated with directed, labeled arcs from heads to dependents. The tree also includes a root node that explicitly marks the head of the entire sentence.

One of the most common ways to illustrate dependency relationships is shown in Figure 1.1 and some of the most common relations are shown in Table 1.2. Dependency parsing is the process of creating trees like this by parsing a sentence, creating the correct arrows and assigning to each one of them the correct label.

Dependency parsing can deal with morphologically rich languages that have a relatively free word order [JM17]. They help us understand the structure and relationships involved in putting words together. For example, dependency parsing can tell us what the subjects and objects of a verb are, as well as which words are modifying/describing the subject. This can be very useful to improve question-answering systems, according to [SL07] and [Cui+05].

Also, Dependency Parsing can be useful in Machine Translation (MT), as [QC06] states, and it has been used to benefit Machine Translation several times in the past. For example, [GM09] used their dependency parser's scores as an extra feature in their machine learning

experiments and they observed significant improvement. According to [Cai+14], pre-ordering rules used in machine translation can be improved by dependency parsing. Finally, in [Yu+15], an evaluation MT metric based on dependency-parsing was created.

1.3 What has been done until now

POS and DP are already supported by toolkits like spaCy[1], NLTK[2] and Stanza[3] in many different languages. However, high performance in these tasks is very important and we can get improvements for a specific language by optimizing the models for that language.

1.4 What we did

For our models we used the GreekBERT transformer [Kou+20]. GreekBERT is an instance of the BERT transformer architecture [Dev+18] which is pretrained on exclusively Greek corpora. A very strong advantage of GreekBERT, as the authors of [Kou+20] point out, other than the extensive Greek datasets on which it was pretrained, is the lower rate of fragmentation in Greek words compared to other multilingual transformer models.

BERT models such as ours use only the encoder transformer modules introduced in [Vas+17] and are pretrained on text corpora in an self-supervised fashion. One of the pre-training tasks is Masked Language Modelling (MLM), in which the model, given a sentence with some words replaced with the MASK token tries to predict the missing words. The other pre-training task is Next Sentence Prediction (NSP), in which the model given two sentences tries to predict if one sentence would follow the other in a document.

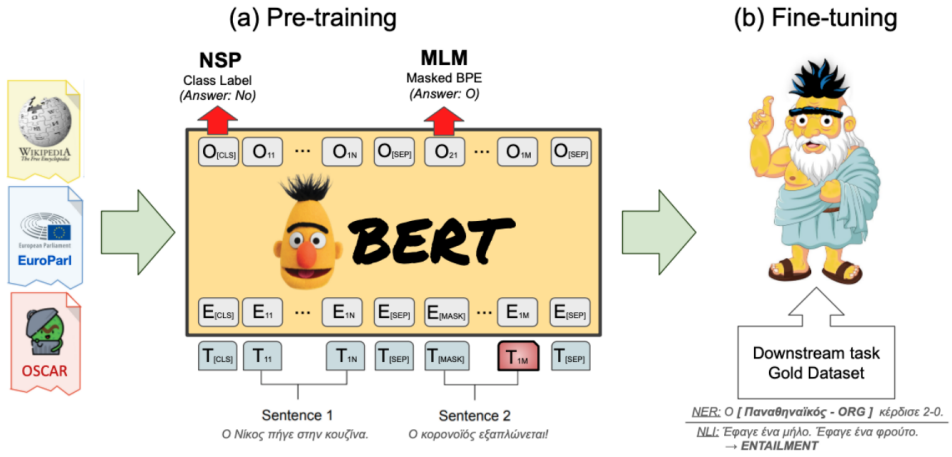
Apart from GreekBERT, for our experiments we also used the XLM-R Transformer [Con+19b]. XLM-R is also an instance of the BERT architecture, but it is pretrained with text sources from multiple languages.

Using GreekBERT or XLM-R, we developed task specific prediction heads for POS tagging and DP and fine-tuned each one to perform well on task specific benchmarks. The benchmark we used is the Greek treebank from the Universal Dependencies dataset. [Niv+16].

We compared our models' performance on the benchmarks mentioned with the performance of the Stanza toolkit [Qi+20] on POS tagging and DP.

[1] https://spacy.io/
[2] https://www.nltk.org/
[3] https://stanfordnlp.github.io/stanza/

Fig. 1.2: An illustration of GreekBERT pretraining. Figure taken from [Kou+20].



Segmentation			
se	##g	##ment	##ation

Tab. 1.3: A word segmented into subword tokens. The initial subword token is "se" and the non-initial sub-word tokens are , ##g , ##ment, ##ation

.

1.5 Handling subwords

Before moving on to the main chapters of the thesis, let us clarify how we use the sub-words (Table 1.3) generated by the tokenizers of the pre-trained Transformer models we employ (GreekBERT, XLM-R). For all tasks we extract labels only for the first subword token of each word and assign these labels to the whole word. However this doesn't mean that the model will not utilize all the sub-words for its predictions. Each layer of GreekBERT or XLM-R, produces embeddings for all the sub-words, and the embedding of each sub-word is a function of the previous layer's embeddings (except the first layer where the embeddings are retrieved from a look-up table) including non initial subword embeddings.

2 Part of Speech Tagging

2.1 Introduction

As already discussed, part of speech tagging is the task of assigning one tag per word in a sentence which contains useful syntactic and semantic information for the word in its context. These tags indicate, for example, if a word is a verb, noun etc.

However there are also more detailed tags that reveal grammatical phenomena that can be assigned to words like the gender of the noun, the degree of an adjective etc. These tags are called morphological tags.

2.2 Related Work

Part of speech tagging has been studied for decades in natural language processing. Here we only briefly mention some indicative previous work. Before Deep Learning emerged, [Kup92] used Hidden Markov Models (HMM) to predict POS tags. [LMP01a] improved POS tagging by introducing Conditional Random Fields (CRFs) [LMP01b], achieving higher accuracy than older HMM models. With the rise of Deep Learning techniques [Col+11] extracted features from text that they fed in a multilayer Feed Forward Neural Network and obtain state-of-the-art results in many tasks, one of which was POS tagging. [DZ14] were the first to use character level information with bidirectional LSTMs [HS97] successfully for POS tagging. [Lin+15] also used a character level LSTM for POS tagging and achieved state of the art results. [NV18] improved the state-of-the-art (at the time) results of UDPipe at POS tagging on the English Penn Treebank. Recently, [Qi+20] reached a state-of-the-art level in many languages for POS tagging and morphological tagging with their multilingual toolkit.

2.3 Dataset

The dataset we used to train and test our model is the Greek treebank from the Universal Dependencies treebank [Niv+16]. This dataset contains data and syntactic annotations for multiple languages. The Greek treebank contains information for every word of the corpus

Sentence	Το	σκορ	του	αγώνα	άνοιξε	ο	Γουέν	Ρούνι	στο	σ	το	22ο	...
Pos tags	DET	X	DET	NOUN	VERB	DET	X	X	_	ADP	DET	NUM	...

Tab. 2.1: An example of a sentence which contains a word with underscore POS tag.

in columns. Every word is annotated with the universal POS tag (UPOS), morhological features (FEATS), the dependant word index (HEAD) as well as the dependency relation (DEPREL).

For fine-grained POS tagging the labels are numerous. We refer the reader to Universal Dependencies website, where complete lists of UPOS tag [1] and morphological features [2] labels exist.

There is also an extra underscore pos tag which is used only in the following pronoun-determiner composites: στο, στα, στην etc. In this case when you encounter these kinds of words in a sentence the two following words are the original composite word divided into its two subwords. Those subwords have the actual part of speech tags (preposition and determiner), while the original composite word does not have a POS tag. For example the word "στο" is divided to σ and το. We used only the two latter subwords to train our model, ignoring the original composite word.

2.4 Properties

We observed some properties and we used them to fairly evaluate our model. The first property is that each part of speech tag has morphological categories from some specific sets of morphological categories. For example, a NOUN cannot have the Voice feature. The second one is that not all words with the same pos tag have the exact same morphological features. For example, NOUN was found with only 3 sets of features, as shown in Table 2.2. Thus, we used the union of all these sets for each pos tag. For instance, given that a word has been predicted as a NOUN we will predict only the labels of the Abbr, Case, Gender and Number features. Tables 2.3 and 2.4 show the sets described above for ADJ and VERV POS tags respectively.

The underscore in Table 2.2 means that some nouns were found with an underscore tag. That is why we added an extra label at each one of the morphological category, an underscore.

[1] https://universaldependencies.org/u/pos/
[2] https://universaldependencies.org/u/feat/index.html

NOUN
Abbr
_
Case, Gender, Number

Tab. 2.2: The sets of morphological categories with which the NOUN POS tag encountered in the training set.

ADJ
Case, Degree, Gender, Number
Case, Gender, Number

Tab. 2.3: The sets of morphological categories with which the ADJ POS tag encountered in the training set

VERB
Aspect, Mood, Number, Person, VerbForm, Voice
Aspect, VerbForm, Voice
Aspect, Mood, Number, Person, Tense, VerbForm, Voice
Aspect, Case, Gender, Number, VerbForm, Voice

Tab. 2.4: The sets of morphological categories with which the VERB POS tag encountered in the training set.

Sentence	Pos tags	Features
Η	DET	'Case': 'Nom', 'Definite': 'Def', 'Gender': 'Fem', 'Number': 'Sing', 'PronType': 'Art'
Μάντσεστερ	X	'Foreign': 'Yes'
Γιουνάιτεντ	X	'Foreign': 'Yes'
ηττήθηκε	VERB	'Aspect': 'Perf', 'Mood': 'Ind', 'Number': 'Sing', 'Person': '3', 'Tense': 'Past', 'VerbForm': 'Fin', 'Voice': 'Pass'
από	ADP	None
την	DET	'Case': 'Acc', 'Definite': 'Def', 'Gender': 'Fem', 'Number': 'Sing', 'PronType': 'Art'
Ατλέτικο	X	'Foreign': 'Yes'
Μπιλμπάο	X	'Foreign': 'Yes'
με	ADP	None
σκορ	X	'Foreign': 'Yes'
2:3	NUM	'NumType': 'Card'

Tab. 2.5: Sentence example 1.

Sentence	Pos tag	Features
Ωστόσο	CCONJ	None
,	PUNCT	None
ο	DET	'Case': 'Nom', 'Definite': 'Def', 'Gender': 'Masc', 'Number': 'Sing', 'PronType': 'Art'
Γουέν	X	'Foreign': 'Yes'
Ρούνι	X	'Foreign': 'Yes'
με	ADP	None
πέναλτι	X	'Foreign': 'Yes'
μείωσε	VERB	'Aspect': 'Perf', 'Mood': 'Ind', 'Number': 'Sing', 'Person': '3', 'Tense': 'Past', 'VerbForm': 'Fin', 'Voice': 'Act'
το	DET	'Case': 'Acc', 'Definite': 'Def', 'Gender': 'Neut', 'Number': 'Sing', 'PronType': 'Art'
σκορ	X	'Foreign': 'Yes'
για	ADP	None
την	DET	'Case': 'Acc', 'Definite': 'Def', 'Gender': 'Fem', 'Number': 'Sing', 'PronType': 'Art'
Μάντσεστερ	X	'Foreign': 'Yes'
.	PUNCT	None

Tab. 2.6: Sentence example 2.

2.5 Model

Our model for POS tagging works as follows: Every input sentence is tokenized and then the GreekBERT or XLM-R model generates embeddings for each token. The model has $N + 1$ linear layers, one for each of the N morphological categories and 1 more for the universal part-of-speech tag. Each one of them takes as input the output embeddings of the bert model. From every layer the class with the highest score is chosen. This method can also be characterised as a multi-task learning approach because the model has many outputs for each input and for each of those outputs a loss is computed. With these losses the model is trained, and over time it learns to predict correct values for every output. For each predicted label we compute a cross-entropy loss. Thus, the final loss is computed by summing all these losses.

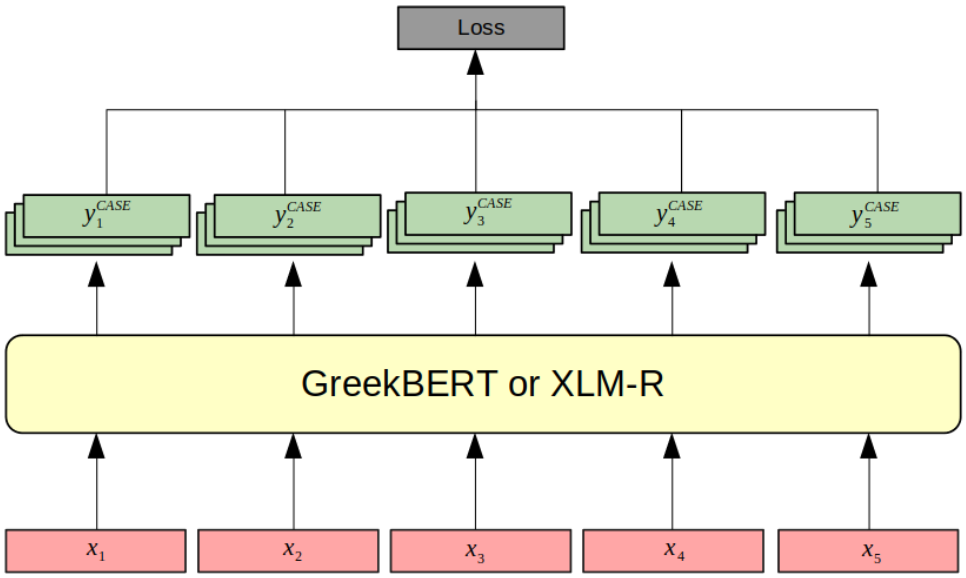
More formally, for input tokens $x_i, i = 0, \ldots, n$, GreekBERT or XLM-R generates embeddings $e_i, i = 0, \ldots, n$, where for every i, $e_i \in \mathbf{R}^{768}$. Finally these embeddings are transformed to $y_{ij} = A_j e_i + b_j \in \mathbf{R}^{C(j)}, j = 0, \ldots, N + 1, i = 0, \ldots, n$ where $C(j)$ is the number of possible classes at the specific output.

2.6 Results

2.6.1 Experimental Results

We calculated f1-scores which is simple because for every output class o and gold class g, a true positive is counted if $o = g$. If $o \neq g$ then a false positive is counted for o and a false negative is counted for g.

Fig. 2.1: An illustration of our POS model. For every input token one prediction is made for every morphological category as well as the universal part-of-speech. The loss is computed by summing the cross-entropy losses for each predicted label.



Learning rate	Dropout	Grad accumulation steps
[5e-5, 3e-5, 2e-5]	[0, 0.1, 0.2]	[4, 8]

Tab. 2.7: Hyperparameters.

We deal with categories and each category contains classes so there are multiple f1-scores for each category. This means that we can describe the performance of each category by micro and macro averages and therefore we have described overall performance of part-of-speech tagging with $N + 1$ numbers. However, because the occurrences of each morphosyntactic category are sparse, the most common category is the underscore so we exclude it from the final scores, in order to emphasize on the valuable information that our model learns.

We tuned the model for the different hyperparameters in Table 3.3 and optimized with grid search the macro-f1 score of the UPOS tag. Early stopping was also used on macro-f1 in order to prevent overfitting. For training we used the AdamW optimizer from PyTorch [LH17] and the cross-entropy loss as the training objective.

We can see that in Table 2.9, that without excluding the underscore class and using the properties, the scores are too high, especially the micro-f1 scores. Using our custom changes to the calculation of the f1-scores will enable us to evaluate the different models more fairly for morphological tagging.

	Simple evaluation		Oversampling		Evaluation using properties	
	micro-f1	macro-f1	micro-f1	macro-f1	micro-f1	macro-f1
POS tag	0.98	0.96	0.98	0.96	0.98	0.96

Tab. 2.8: Micro-f1 and macro-f1 results for POS tagging.

Morphosyntactic category	Simple evaluation		Evaluation using properties	
	micro-f1	macro-f1	micro-f1	macro-f1
Case	0.98	0.99	0.98	0.97
Definite	1	1	1	1
Gender	0.99	0.98	0.98	0.98
Number	0.99	0.99	0.99	0.99
PronType	1	0.98	1	0.97
Foreign	0.99	0.88	0.88	0.88
Aspect	1	0.99	0.99	0.99
Mood	1	0.96	0.99	0.83
Person	1	1	1	1
Tense	1	1	1	1
VerbForm	1	0.96	0.99	0.93
Voice	1	0.97	0.96	0.96
NumType	1	0.95	0.99	0.96
Poss	1	1	0.98	0.98
Degree	1	0.75	0.86	0.50
Abbr	1	0.97	0.94	0.94

Tab. 2.9: Micro-f1 and macro-f1 results using plain f1 score calculation and evaluation using properties.

Metric	Stanza	Oversampled	Regular
micro-f1 tag	0.98	0.98	0.98
macro-f1 tag	0.96	0.97	0.97

Tab. 2.10: Micro-f1 and macro-f1 results for Stanza, the oversampled model and the regular model.

Metric	GreekBERT	XLMR
micro-f1	0.98	0.98
macro-f1	0.97	0.97

Tab. 2.11: Micro-f1 and macro-f1 results for GreekBERT and XLMR, results on the test set.

However, we can easily see that most morphosyntactic categories achieve respectable micro-f1 scores but there are some with especially bad macro-f1 scores. For the Degree and Mood categories we have found that some of their classes have very few occurrences in the training set. In order to solve this problem we oversampled in the training set any sentences that contained words in the problematic classes of the aforementioned categories. The results show that there is a significant improvement in the macro-f1 scores for the problematic categories.

Moreover, we use the properties described in the Properties section so given the pos tag we evaluate our model using only the expected morphological features.

2.6.2 Comparing to Stanza

We ran the Stanza [Qi+20] NLP toolkit to the same test set we used which contains 456 sentences. However, we excluded 20 sentences because the tokenization was different from the expected. One of the sentences has the problem that the word 'E.E.' is split to 'E.E' and '.' while the other 19 have the same problem that only in some cases before words like στις, στα etc an extra capital σ, Σ is added. We computed the f1 scores using the properties we describe in Section 2.4 section.

In tables 2.10 and 2.11 the results show that for predicting POS tags the models' performance is almost the same and in tables 3.3 and 2.13 the results show that for predicting morphological tags the models perform very similarly in all categories.

Morphosyntactic category	GreekBERT		XLMR	
	micro-f1	macro-f1	micro-f1	macro-f1
Case	0.98	**0.97**	0.98	0.78
Definite	1	1	1	1
Gender	**0.98**	**0.98**	0.96	0.95
Number	0.99	0.99	0.99	0.99
PronType	1	**0.97**	0.99	0.69
Foreign	0.88	0.88	**0.91**	**0.91**
Aspect	0.99	0.99	0.98	0.98
Mood	0.99	**0.83**	0.99	0.59
Person	1	1	0.99	0.99
Tense	1	1	0.99	0.99
VerbForm	0.99	0.93	1	**0.97**
Voice	0.96	0.96	0.98	0.98
NumType	0.99	**0.96**	0.93	0.70
Poss	0.98	0.98	0.97	0.97
Degree	0.86	0.50	0.88	0.47
Abbr	0.94	0.94	0.95	0.95

Tab. 2.12: Comparison of mophosyntactic category scores using XLMR and GreekBERT to generate embeddings, results on test set.

Morphosyntactic category	Stanza toolkit		Oversampled model		Regular model	
	micro-f1	macro-f1	micro-f1	macro-f1	micro-f1	macro-f1
Case	0.98	0.97	0.98	0.95	0.98	0.97
Definite	1	1	1	1	1	1
Gender	0.97	0.97	0.98	0.98	0.98	0.98
Number	0.99	0.99	0.99	0.99	0.99	0.99
PronType	0.99	0.94	1	0.95	1	0.97
Foreign	0.79	0.79	**0.88**	**0.88**	**0.88**	**0.88**
Aspect	0.98	0.98	0.99	0.98	0.99	0.99
Mood	0.99	0.59	1	**0.97**	0.99	0.83
Person	0.99	0.98	1	0.99	1	1
Tense	0.98	0.98	1	1	1	1
VerbForm	1	0.97	1	0.96	0.99	0.93
Voice	**0.99**	**0.99**	0.97	0.96	0.96	0.96
NumType	0.95	0.93	0.94	0.93	0.99	0.96
Poss	0.96	0.96	0.99	0.99	0.98	0.98
Degree	0.89	0.57	0.96	**0.87**	0.86	0.50
Abbr	0.96	0.96	0.96	0.96	0.94	0.94

Tab. 2.13: Micro-f1 and macro-f1 results for Stanza, the Oversampled model and the Regular model as well as XLMR (not oversampled).

3 Dependency Parsing

3.1 Problem statement

As already discussed, dependency parsing is the task of predicting which word depends on which from a sentence. This is formalised as extracting a dependency graph in a sentence. Each edge (i, j) in this graph means that word i is dependant on word j. For (i, j) we also call j the head of the word i. We represent this relationship with an arrow pointing from the head to the dependant word. Every arc also has one label describing the way word i modifies word j.

3.1.1 Related Work

Like POS tagging, dependency parsing has been studied for decades in natural language processing. Here we briefly refer to only some indicative previous work. [Eis97] developed the first statistical models that were based on context-free grammars (CFG) and used graph-based parsing but only allowing projective dependency graphs. [McD+05] formalize the problem of predicting the dependency graph given arc scores as finding the Minimum Spanning Tree of the weighted graph but extended it to non-projective dependency graphs allowing for an increase in accuracy in non-projective languages. When Deep Learning emerged, [KG16] were the first to use LSTMs for DP to the best of our knowledge. More recently, [DQM17] used biaffine classifiers to produce the arc scores from LSTM embeddings achieving state-of-the art in multiple languages.

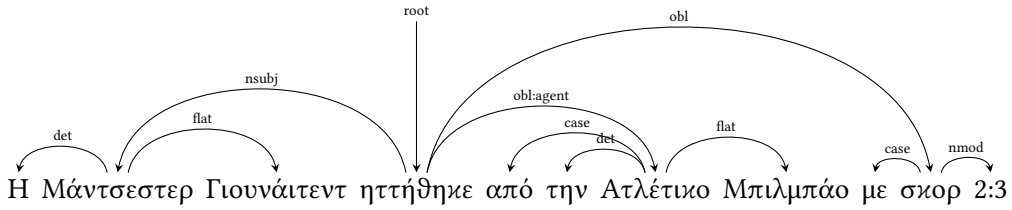


Fig. 3.1: Dependency tree example 1.

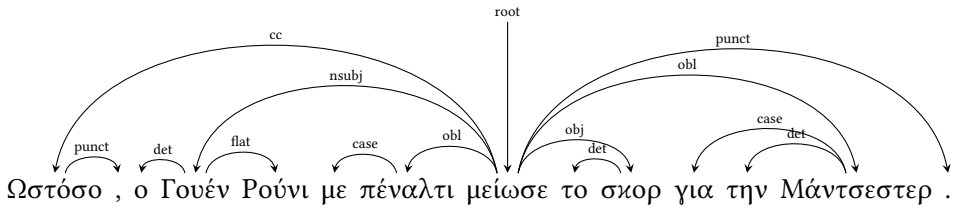


Fig. 3.2: Dependency tree example 2.

Index	Sentence	Head	Deprel
1	Η	2	det
2	Μάντσεστερ	4	nsubj
3	Γιουνάιτεντ	2	flat
4	ηττήθηκε	0	root
5	από	7	case
6	την	7	det
7	Ατλέτικο	4	obl:agent
8	Μπιλμπάο	7	flat
9	με	10	case
10	σκορ	4	obl
11	2:3	10	nmod

Tab. 3.1: Sentence example 1.

3.2 Dataset

The dataset we use is the same as in Section 2 with the only difference being that the fields of interest are the head of each word and the label of the arc to the head. Many authors have used the existing POS tag information of the dataset to obtain more features and information for dependency parsing and others have predicted jointly the dependency graph and the POS tags of each word. We explore the multitask case later on.

More specifically, the head of each word is called 'head' in the dataset and it is represented by a number which indicates the dependant word index in the sentence while the label of the arc to the head is called 'deprel' (dependency relation) in the dataset and it indicates the label of the relation between the specific word and the dependant one.

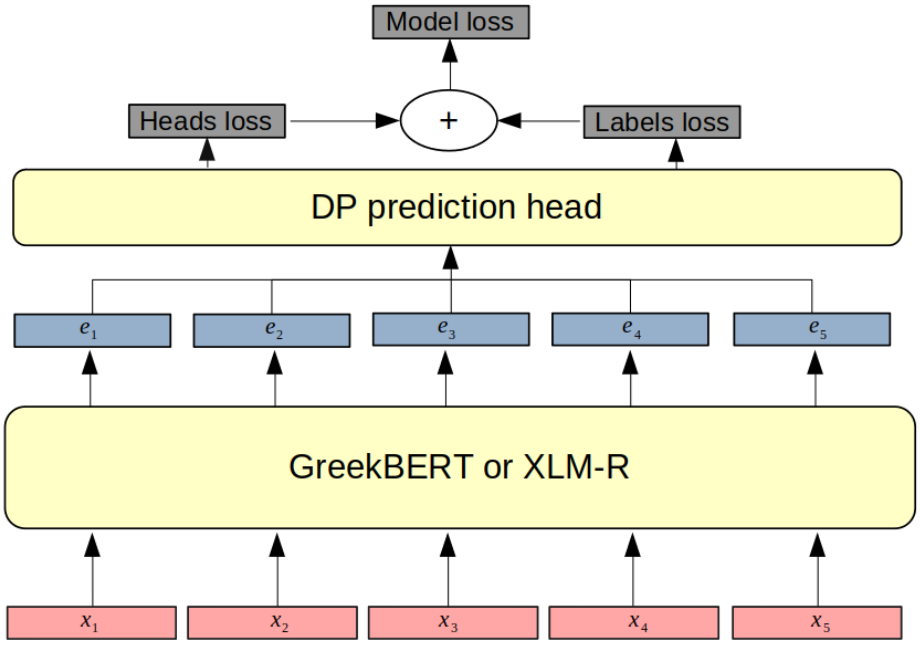
The indexing of real words in all sentences starts at 1. However, there is a virtual node which is called root and is indexed with 0. There is one node with the root dependency relation in every tree and is called the root word. It is not dependant on any other word and it is usually the main verb of the sentence.

Word Index	Sentence	Head	Deprel
1	Χθες	6	advmod
2	,	1	punct
3	η	4	det
4	Μάντσεστερ	6	nsubj
5	Γιουνάιτεντ	4	flat
6	ηττήθηκε	0	root
7	με	8	case
8	σκορ	6	obl
9	2:3	8	nmod
10	από	12	case
11	την	12	det
12	Ατλέτικο	6	obl:agent
13	Μπιλμπάο	12	flat
14	,	17	punct
15	στα	None	_
16	σ	17	case
17	τα	17	det
18	πλαίσια	6	obl
19	της	19	det
20	φάσης	17	nmod
21	των	21	det
22	16	19	nmod
23	του	23	det
24	Γιουρόπα	21	nmod
25	Λιγκ	23	flat
26	2011-2012	23	nmod
27	.	6	punct

Tab. 3.2: Sentence example 2.

3.3 Model

Fig. 3.3: An illustration of the dependency parsing model. Tokens are fed into our transformer model which produces embeddings. With these embeddings arc and label predictions are made. A loss is calculated for the arcs as well as the labels which are both summed to produce a total model loss.



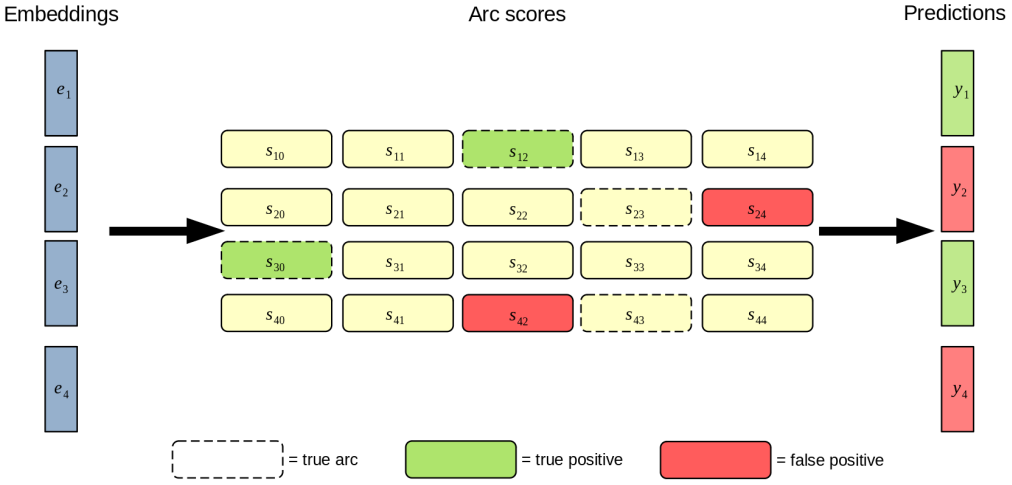
The model outputs one score for each possible arc in the graph. Essentially, the arc score for every arc (i, j), where i, j are words in the sentence, is calculated as $e_i W e_j^T$ where e_i, e_j are the transformer's embeddings for words i and j respectively.

After the arc scores have been calculated we take the best scoring arc starting from i. Formally we select $y_i = arg\,max_j scores(i, j)$. With the predicted arcs the model can now predict the labels of each arc.

More concretely, the following representations are created from the embeddings:

$$
\begin{aligned}
h_i^{(arc-dep)} &= W^{(arc-dep)} e_i \\
h_i^{(arc-head)} &= W^{(arc-head)} e_i \\
h_i^{(rel-dep)} &= W^{(rel-dep)} e_i \\
h_i^{(rel-head)} &= W^{(rel-head)} e_i
\end{aligned}
$$

Fig. 3.4: An illustration of the arc scoring mechanism. For every pair of input tokens a score is calculated, then the arc starting at each token with the highest score is selected.



The matrices $W^{(arc-dep)}, W^{(arc-head)}, W^{(rel-dep)}, W^{(rel-head)}$ are parameters of the model and they are learned. and their sizes are 768x768 meaning that the produced representations have the same dimensions as the original embeddings.

These representations also have intuitive meanings. $h_i^{(arc-dep)}$ is the representation of token i as a head seeking its dependant and $h_i^{(arc-head)}$ is the representation of token i as a dependant looking for its head.

We then produce scores for each possible arc (i, j) :

$$s_{ij}^{(arc)} = (h_j^{(arc-head)})^T W^{(arc)} h_i^{(arc-dep)} + (h_j^{(arc-head)})^T b^{(arc)}$$

Here, $W^{(arc)}$ and $b^{(arc)}$ are parameters that are learned and have size 768x768 and 768 respectively. These terms have intuitive interpretations. The first term is the probability of an arc (i, j) existing given that the dependant is word i and the head word j and the second term is the probability of an arc (i, j) existing only given that the head is word j.

With these scores we can then predict the head for each token x_i as the receiving end of the arc (i, j) with the highest score.

$$y_i^{(arc)} = \arg\max_j s_{ij}^{(arc)}$$

This can also be thought as a sequence-labeling problem because we need to predict one label (the head) for each token.

For the label scores we generate for each arc (i, j), K scores, one for each possible dependency relation. (Note that $\oplus$ means concatenation.)

$$s_{ijk}^{(rel)} = (h_j^{(rel-head)})^T U_k^{(rel)} h_i^{(rel-dep)} + w_k^T (h_j^{(rel-head)} \oplus h_i^{(rel-dep)}) + b_k^{(rel)}$$

The intuitive interpretation here is that the first term is the probability of the label being k given that the head is j and the dependant i. The second term is the probability of label k given either the head being j or the dependant being i and finally the last term is the prior probability of a label being k.

The difference comparing to the arc scores is that a linear layer is used for the labels and we need essentially K outputs for the bilinear layer, which is achieved with the K different U_k^{rel} matrices. The parameters here are the K U_k^{rel} matrices each of size 768x768, the K w_k vectors of size 1536x1 and the bias terms $b = b_k^{(rel)}$.

The predictions then for the labels use the predictions for the arcs.

$$y_i^{(rel)} = \arg\max_k s_{iy_i^{(arc)}k}^{(rel)}$$

For each of these output scores we can calculate a cross-entropy loss when we have the true labels available. These losses are then summed which produces a total model loss.

$$\mathcal{L}(x_i) = \mathcal{L}_{\text{ARC}}(x_i) + \mathcal{L}_{\text{REL}}(x_i)$$

3.4 Results

3.4.1 Experimental results

The metrics used in dependency parsing are Unlabeled Attachement Score (UAS) and Labeled Attachment Score (LAS). UAS is the amount of words that got assigned to the correct head divided by the total amount of words. LAS is the amount of words that got assigned the correct head and the correct dependency relation divided by the total amount of words. In other words, UAS is the percentage of the words that get the correct head,

Learning rate	Dropout	Grad accumulation steps
[5e-5, 3e-5, 2e-5]	[0, 0.1, 0.2]	[4, 8]

Tab. 3.3: Hyperparameters.

	Stanza	GreekBERT	XLM-R
UAS	0.91	**0.94**	0.89
LAS	0.88	**0.92**	0.87

Tab. 3.4: UAS and LAS scores in the test set for Stanza, our GreekBERT based model and our XLM-R based model.

while the LAS is the percentage of the words that get both the correct head and label. So UAS is independent of label, while LAS considers both.

As already discussed, we experimented with two pre-trained Transformer-based models for producing the embeddings, GreekBERT and XLM-R. We tuned the models for the different hyperparameters in 3.3 and optimized with grid search the UAS score. Early stopping was also used on UAS in order to avoid overfitting. For training we used the AdamW optimizer from PyTorch [LH17] and the cross-entropy loss as the training objective.

3.4.2 Comparing to stanza

We ran the Stanza [Qi+20] NLP toolkit to the same test set we used. However, we have excluded 20 sentences because the tokenization was different from the expected one as we explained thoroughly in Section 2.6.2.

From the results in Table 3.4 we see that our parser with the GreekBERT outperforms both Stanza and our XLM-R based parser by a large margin.

4 Conclusions and Future work

4.1 Key takeaways

We developed a model that can do POS and DP that is optimized for the Greek language. The models achieve state-of-the-art results in all tasks. Significant gains in performance were observed in DP, where our model outperformed Stanza's Greek dependency parser by a large margin. Also, XML-R performed much worse than GreekBert in DP. We believe that this gain in performance can be largely attributed to the extensive Greek datasets on which GreekBERT was pretrained.

4.2 Further work

This Greek NLP pipeline can be extended with many more tasks. First of all, toxicity detection can be added. Toxicity detection is very important for use cases where users can comment or review a certain product. In these cases the inputs from the users need to be moderated in case they contain harmful language [Pav+17], [Pav+20], [Pav+21].

Making the pipeline more efficient is also very important not only for the environmental cost of the models but also for speeding up systems. Distillation [HVD15] and Quantization [Hub+17] are widely used techniques especially in BERT models. A user could theoretically choose some amount of accuracy loss in return for speed. This could further be combined with Multi-task Learning (MTL) for more speed up gains.

Finally, it's good to have highly tuned models for these tasks because the users of the features of the toolkit will benefit greatly from an increase in performance. There are many parameters that can be tuned more and alternative learning strategies. For example for hyperparameter tuning, Hyperopt [BYC+13] can be used to also utilize possible hyperparameters that are continuous such as learning rate, which we tuned using Grid Search.

Bibliography

[BYC+13] James Bergstra, Dan Yamins, David D Cox, et al. "Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms". In: *Proceedings of the 12th Python in science conference*. Vol. 13. Citeseer. 2013, p. 20.

[Cai+14] Jingsheng Cai, Masao Utiyama, Eiichiro Sumita, and Yujie Zhang. "Dependency-based pre-ordering for Chinese-English machine translation". In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 2014, pp. 155–160.

[CM98] Abdur Chowdhury and M Catherine McCabe. *Improving information retrieval systems using part of speech tagging*. Tech. rep. 1998.

[Col+11] Ronan Collobert, Jason Weston, Léon Bottou, et al. "Natural language processing (almost) from scratch". In: *Journal of machine learning research* 12.ARTICLE (2011), pp. 2493–2537.

[Con+19a] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, et al. "Unsupervised cross-lingual representation learning at scale". In: *arXiv preprint arXiv:1911.02116* (2019).

[Con+19b] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, et al. "Unsupervised cross-lingual representation learning at scale". In: *arXiv preprint arXiv:1911.02116* (2019).

[Cui+05] Hang Cui, Renxu Sun, Keya Li, Min-Yen Kan, and Tat-Seng Chua. "Question answering passage retrieval using dependency relations". In: *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*. 2005, pp. 400–407.

[De +14] Marie-Catherine De Marneffe, Timothy Dozat, Natalia Silveira, et al. "Universal Stanford dependencies: A cross-linguistic typology." In: *LREC*. Vol. 14. 2014, pp. 4585–4592.

[Dev+18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[DQM17] Timothy Dozat, Peng Qi, and Christopher D Manning. "Stanford's graph-based neural dependency parser at the conll 2017 shared task". In: *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. 2017, pp. 20–30.

[DZ14] Cicero Dos Santos and Bianca Zadrozny. "Learning character-level representations for part-of-speech tagging". In: *International Conference on Machine Learning*. PMLR. 2014, pp. 1818–1826.

[Eis97] Jason Eisner. "Three new probabilistic models for dependency parsing: An exploration". In: *arXiv preprint cmp-lg/9706003* (1997).

[Far+08] Benjamin Farber, Dayne Freitag, Nizar Habash, and Owen Rambow. "Improving NER in Arabic Using a Morphological Tagger." In: *LREC*. 2008.

[GM09] Michel Galley and Christopher D Manning. "Quadratic-time dependency parsing for machine translation". In: *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*. 2009, pp. 773–781.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[Hub+17] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. "Quantized neural networks: Training neural networks with low precision weights and activations". In: *The Journal of Machine Learning Research* 18.1 (2017), pp. 6869–6898.

[HVD15] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. "Distilling the knowledge in a neural network". In: *arXiv preprint arXiv:1503.02531* (2015).

[JM17] Dan Jurafsky and James H. Martin. *Speech and Language Processing (3rd ed. Draft)*. 2017.

[KG16] Eliyahu Kiperwasser and Yoav Goldberg. "Simple and accurate dependency parsing using bidirectional LSTM feature representations". In: *Transactions of the Association for Computational Linguistics* 4 (2016), pp. 313–327.

[Kou+20] John Koutsikakis, Ilias Chalkidis, Prodromos Malakasiotis, and Ion Androutsopoulos. "Greek-bert: The greeks visiting sesame street". In: *11th Hellenic Conference on Artificial Intelligence*. 2020, pp. 110–117.

[Kup92] Julian Kupiec. "Robust part-of-speech tagging using a hidden Markov model". In: *Computer speech & language* 6.3 (1992), pp. 225–242.

[LH17] Ilya Loshchilov and Frank Hutter. "Decoupled weight decay regularization". In: *arXiv preprint arXiv:1711.05101* (2017).

[Lin+15] Wang Ling, Tiago Luıs, Luıs Marujo, et al. "Finding function in form: Compositional character models for open vocabulary word representation". In: *arXiv preprint arXiv:1508.02096* (2015).

[LMP01a] John Lafferty, Andrew McCallum, and Fernando CN Pereira. "Conditional random fields: Probabilistic models for segmenting and labeling sequence data". In: (2001).

[LMP01b] John Lafferty, Andrew McCallum, and Fernando CN Pereira. "Conditional random fields: Probabilistic models for segmenting and labeling sequence data". In: (2001).

[McD+05] Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajic. "Non-projective dependency parsing using spanning tree algorithms". In: *Proceedings of human language technology conference and conference on empirical methods in natural language processing*. 2005, pp. 523–530.

[Niv+16] Joakim Nivre, Marie-Catherine De Marneffe, Filip Ginter, et al. "Universal dependencies v1: A multilingual treebank collection". In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. 2016, pp. 1659–1666.

[NV18] Dat Quoc Nguyen and Karin Verspoor. "An improved neural network model for joint POS tagging and dependency parsing". In: *arXiv preprint arXiv:1807.03955* (2018).

[Pav+17] John Pavlopoulos, Prodromos Malakasiotis, Juli Bakagianni, and Ion Androutsopoulos. "Improved abusive comment moderation with user embeddings". In: *arXiv preprint arXiv:1708.03699* (2017).

[Pav+20] John Pavlopoulos, Jeffrey Sorensen, Lucas Dixon, Nithum Thain, and Ion Androutsopoulos. "Toxicity Detection: Does Context Really Matter?" In: *arXiv preprint arXiv:2006.00998* (2020).

[Pav+21] John Pavlopoulos, Léo Laugier, Jeffrey Sorensen, and Ion Androutsopoulos. "Semeval-2021 task 5: Toxic spans detection". In: *Proceedings of SemEval* (2021).

[QC06] Chris Quirk and Simon Corston-Oliver. "The impact of parse quality on syntactically-informed statistical machine translation". In: *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*. 2006, pp. 62–69.

[Qi+20] Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D Manning. "Stanza: A Python natural language processing toolkit for many human languages". In: *arXiv preprint arXiv:2003.07082* (2020).

[SB11] Ming Sun and Jerome R Bellegarda. "Improved pos tagging for text-to-speech synthesis". In: *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2011, pp. 5384–5387.

[SL07] Dan Shen and Mirella Lapata. "Using semantic roles to improve question answering". In: *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*. 2007, pp. 12–21.

[Smy21] Nikolaos Smyrnioudis. "A Transformer-based Natural Language Processing Toolkit for Greek – Named Entity Recognition and Multi-task Learning Experiments". 2021.

[Vas+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, et al. “Attention is all you need”. In: *Advances in neural information processing systems*. 2017, pp. 5998–6008.

[Yu+15] Hui Yu, Xiaofeng Wu, Wenbin Jiang, Qun Liu, and Shouxun Lin. “An automatic machine translation evaluation metric based on dependency parsing model”. In: *arXiv preprint arXiv:1508.01996* (2015).

List of Acronyms

NLP Natural Language Processing

NER Named Entity Recognition

POS Part of speech tagging

DP Dependency parsing

MTL MultiTask Learning

MLM Masked Language Modelling

NSP Next Sentence Prediction

List of Figures

List of Tables