Towards Joint Morphological Analysis and Dependency Parsing of Turkish

Özlem Çetinoğlu and **Jonas Kuhn**
IMS, University of Stuttgart
Germany
{ozlem,jonas}@ims.uni-stuttgart.de

Abstract

Turkish is an agglutinative language with rich morphology-syntax interactions. As an extension of this property, the Turkish Treebank is designed to represent sublexical dependencies, which brings extra challenges to parsing raw text. In this work, we use a joint POS tagging and parsing approach to parse Turkish raw text, and we show it outperforms a pipeline approach. Then we experiment with incorporating morphological feature prediction into the joint system. Our results show statistically significant improvements with the joint systems and achieve the state-of-the-art accuracy for Turkish dependency parsing.

1 Introduction

Turkish is a morphologically rich language (MRL) that has been known to pose interesting research questions to linguists and computational linguists, including architectural issues at the morphology-syntax interface. Today, good quality tools for morphological analysis are available for analysing Turkish raw text input at the word level, and in work on the Turkish Dependency Treebank (Oflazer et al., 2003), a representation scheme has been developed that captures the peculiarities at the morphology-syntax interface in a dependency format that is formally compatible with the standard CoNLL dependency format.

So, it might seem as if all Turkish-specific challenges have been resolved, and only language-independent data-driven methods are required from now on (after all, the Turkish Dependency Treebank was included in the CoNLL 2006 and 2007 Shared Tasks (Buchholz and Marsi, 2006; Nivre et al., 2007), and several researchers working on language-independent methods have reported scores on the available data).

However, Turkish still causes a considerable architectural challenge for the standard pipeline architecture used in data-driven dependency parsing: the dependency treebank scheme for Turkish is based on segments that are not identical to the words from the raw text input, but are often sublexical units that form parts of morphological derivations.

While it is straightforward to train data-driven parsers on the gold standard segmentation from the treebank (which is what happened in the shared tasks), any realistic application starting out with raw text has to involve morphological disambiguation in the preprocessing which means it is not guaranteed that treebank-compatible segment boundaries will be produced. For instance, when training a dependency parser on predicted POS and morphology features, the treebank is of course used to provide the gold standard dependency arcs, but with an automatic (and hence imperfect) morphological disambiguator, there will be cases where the gold standard assumes two segments for a word, but morphological prediction assumes only one. So any standard learning algorithm will break down because the node sets for the dependency graphs are incompatible.

For many languages, realistic parsing scenarios assume gold tokens and use predicted POS (and morphological features). For Turkish, keeping the gold segmentation and assigning predicted POS and morphology would converge to using an oracle because gold segmentation would sometimes disambiguate morphology. Instead, realistic scenarios include segmentation, and a statistical morphological disambiguator picks the most probable analysis among all possibilities a morphological analyser produces. It is the morphological analysis that determines the lemma, POS, morphological features, and segmentation of a word is based on the number of its word-internal derivations.

For instance, in (1), the middle word *bende* has

Proceedings of the Second International Conference on Dependency Linguistics (DepLing 2013), pages 23–32,
Prague, August 27–30, 2013.

four morphological analyses with different lemma and POS combinations, meaning 'at me', 'on the mole', 'to the dam', and 'servant' respectively.[1] Hence, unlike many other languages, the segmentation, POS tagging, and morphological analysis are tightly connected for Turkish.

Eryiğit et al. (2008) is the first work that addresses the segmentation problem in parsing predicted text. They set up a pipeline architecture of a morphological analyser and disambiguator but leave out handling the multiword expressions.[2] A recent work from Eryiğit (2012) solely focuses on the impact of the morphological analysis and disambiguation of the Turkish treebank. Again, it follows the standard pipeline but this time with a treebank version that represents multiwords as detached segments, which allows avoiding to use a multiword extractor.

The major drawback of a pipeline system is to propagate the disambiguator's mistakes to the parsing step. Moreover, the disambiguator cannot take advantage of syntactic information that could help disambiguate certain morphological analyses.

In (1), the first word *kahveleri* means 'the coffees (Acc)', 'his/her coffees', 'their (one) coffee', 'their coffees' from (1a) to (1d). When the first two words come together, they make a sentence meaning 'His/her/their coffees are at my place'. *kahveleri* is still ambiguous but its dependency relation is clear; *bende*, with morphological analysis (1e), behaves as a copular predicate with no overt marker and *kahveleri* is dependent on *bende* as a subject.

When the third word *içelim* 'let's drink' follows the former two, the meaning of the sentence changes to 'Let's drink the coffees at my place', which also changes the morphological analysis of *kahveleri* to (1a). It now behaves as the object of the main predicate *içelim*. A pipeline system cannot benefit from such a disambiguation advantage.

An alternative approach to pipeline architectures is making joint decisions on morphological disambiguation and parsing. It has been shown that such an architecture improves constituency parsing accuracy both for Arabic (Green and Manning, 2010) and for Hebrew (Goldberg and Tsarfaty, 2008). On the dependency parsing front, Lee et al. (2011) introduces a joint morphological disambiguation and dependency parsing architecture which proves to outperform their pipeline architecture for Latin, Ancient Greek, Czech, and Hungarian. However it is limited to unlabelled dependency parsing and initial scores are below the state-of-the-art. On the other hand, parsers that can jointly POS tag become more common in the last years (Bohnet and Nivre, 2012; Hatori et al., 2011; Li et al., 2011). Bohnet and Nivre (2012) propose a joint POS tagger and labelled dependency parser that outperforms the pipeline results and also improves the state-of-the-art accuracy for German, Czech, English, and Chinese.

Joint POS tagger and dependency parsers are not originally designed for predicting morphological features, but they provide a flexible field (POS) where the parser is not dependent on the morphological disambiguator decisions. So the use of this field can actually be extended to accommodating morphological features instead of or in addition to POS tags, which gives parsers an opportunity to override fixed disambiguator mistakes. Hence, those parsers approximate to a joint morphological disambiguation and dependency parsing architecture, which provides us with a testbed until genuinely full-fledged joint parsers are developed.

In this paper we use Bohnet and Nivre's (2012) system to apply their approach to Turkish and later to explore ways to include morphological feature prediction into parsing. Experimental results show that even a partial flexibility in predicting the morphological features helps improve the parsing accuracy statistically significantly.

The paper is structured as follows: Section 2 gives an overview on how morphological features are used in parsing MRLs. Section 3 explains the morphological analysis representation and its relation with segmentation. Section 4 describes the use of morphological features in joint parsing experiments. The setup for experiments are given in Section 5 and results are discussed in Section 6. We conclude with Section 7.

2 Use of Morphological Features

Using morphological information as features in parsing has been a commonly used method for MRLs (Tsarfaty et al., 2010). The effect is controversial: in some cases gold morphology clearly

[1] A3pl: 3rd personal plural agreement, A3sg: 3rd personal singular agreement, Pnon: no possessives, P3sg: 3rd personal singular possessive, P3pl: 3rd personal plural possessive, Nom: Nominative, Acc: Accusative, Loc: Locative, Dat: Dative, Zero: No overt derivation, Pos: Positive, Opt: Optative mood

[2] because of the lack of a multiword extractor. Hence the experiments are not in a fully predicted setting.

	Kahveleri	bende	içelim
(1)	a.kahve+Noun+A3pl+Pnon+Acc	e.ben+PersP+A1sg+Pnon+Loc	iç+Verb+Pos+Opt+A1pl
	b.kahve+Noun+A3pl+P3sg+Nom	f.ben+Noun+A3sg+Pnon+Loc	
	c.kahve+Noun+A3sg+P3pl+Nom	g.bent+Noun+A3sg+Pnon+Dat	
	d.kahve+Noun+A3pl+P3pl+Nom	h.bende+Noun+A3sg+Pnon+Nom	

helps, in others its impact is little. For some settings predicted information causes a drop, for some settings a partial set of morphological features improves parsing accuracy.

Ambati et al. (2010) explore ways of integrating local morphosyntactic features into Hindi dependency parsing. They experiment with different sets of features both on a graph-based and a transition-based dependency parser. Both with gold and predicted settings using morphological features root, case, and suffix outperform using POS as the only feature.

Bengoetxea and Gojenola (2010) utilise the CoNLL-X format and MaltParser's feature configuration file to take advantage of morphological features in parsing Basque with gold data. Their experiments show that case and subordination type increase parsing accuracy.

Marton et al. (2010) explore which morphological features could be useful in dependency parsing of Arabic. They observe the effect of features by adding them one at a time separately and comparing the outcomes. Experiments show that when gold morphology is provided, case markers help the most, whereas when the morphology is automatically predicted the outcome is the opposite: using case harms the results the most. When features are combined in a greedy heuristic, using definiteness, person, number, and gender information improves accuracy.

To overcome the exhaustive feature space problem of Arabic, Dehdari et al. (2011) use heuristic search algorithms for the optimal feature combination. Similar to Marton et al. (2010) they run experiments by including one feature at a time to their no-feature baseline, and also conduct a second set of experiments where they remove one feature at a time from the whole feature set. They also conclude that leaving out the predicted case improved the parsing most among the possible candidates to remove, this time for constituency parsing. In the single feature experiments, genitive clitics help the most. The optimal combination they achieve consists of the features determiner, proper noun, genitive clitics, and negation.

Another Semitic language that is studied within the MRLs is Hebrew. Initial results on Hebrew dependency parsing (Goldberg and Elhadad, 2009) show predicted morphological features help in a transition-based parser with a tailored feature configuration file, although scores drop in a graph-based parser. The same authors later prove both gold and predicted agreement features improve accuracy for an easy-first, non-directional dependency parser (Goldberg and Elhadad, 2010). Tsarfaty and Sima'an (2010) report agreement features are useful also for constituency parsing when they extend the Relational Realisational (Tsarfaty and Sima'an, 2008) models with this information.

Seeker and Kuhn (2011) focus on the internal structures and grammatical functions of German noun phrases. Their experiments show grammatical functions are predicted with higher accuracy when a graph-based dependency parser is provided with both gold and predicted case markers.

They further explore the effects of using case in dependency parsing, this time for Czech and Hungarian as well as for German (Seeker and Kuhn, 2013). On a graph-based parser German does not benefit much from using predicted morphology but Czech and Hungarian clearly profit. They also use case as a constraint on integer linear programming (ILP) parsing models to filter out ungrammatical case-function mappings. For all three languages, the constrained models outperform the unconstrained models and graph-based parser in predicting core grammatical functions.

The research discussed in this section show case and agreement are among the most investigated features, and most of the time they are among the most beneficial ones. These are the features we also look into. But first, we describe the interaction between the morphology and syntax in Turkish in Section 3.

3 The Morphology-Syntax Interface in Turkish

The motivation behind using sublexical units in the Turkish treebank comes from its agglutinative nature. Many linguistic phenomena that are

syntactic in other languages are represented with derivational morphology in Turkish (Sulger et al., 2013). For instance *çekti* is a one-word sentence in Turkish meaning 'It was a cheque'. The word *çek* 'cheque' is derived into a verb (with no overt suffix) and then the past tense suffix *-ti* is attached. (2) is the morphological representation of this word where `^DB` denotes the derivational boundary:

(2) `çek+Noun+A3sg+Pnon+Nom^DB+Verb+Zero +Past+A3sg`

Each sequence of inflectional features divided by a derivational boundary is called an *inflectional group* (IG hereafter). The word in (2) has two IGs. A further example clarifies why inflectional groups are chosen as the unit of the treebank. Figure 1 gives the dependency representation of the sentence *açık çekti* 'It was a blank cheque'. The adjective *açık* 'blank' modifies the noun *çek* only, not the derived verb *çekti*. A word based representation would disregard this distinction.

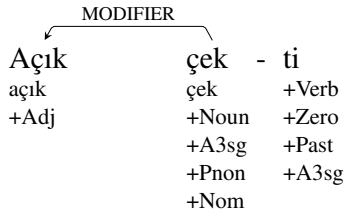

Figure 1: The dependency representation for *açık çekti*

The Turkish Treebank follows this IG notation. A word is segmented into segments from its derivational boundaries. If it is derived n times, it is represented as n+1 segments. The first segment has the lemma, and the last segment has the whole word as the surface form. The surface forms of non-final segments are underscores. (3) gives the treebank representation of the sentence in Figure 1 in the CoNLL format. The derived verb *çekti* is represented as two segments.

The possible segmentation problem arises when words have ambiguous morphological analyses with different number of IGs. For instance, the word *çekti* has a second interpretation with the meaning 's/he pulled' which is the past tense of the verb *çek* 'to pull' in 3rd person singular. The morphological representation of this sense is given in (4).

(4) `çek+Verb+Pos+Past+A3sg`

Note that in this analysis, there are no derivational boundaries, hence it only has a single IG. When the gold standard is the first interpretation of the word *çekti* and a morphological prediction suggests the second interpretation, the number of segments do not match any more.

4 Morphological Feature Prediction

Like many other free-word-order languages, Turkish has overt case markers. It is the case marker that determines the function of a word in a sentence rather than the POS of that word. For instance, an accusative nominal is an object no matter if it is a noun, proper noun, or pronoun.

However, the case-function mapping is not completely unambiguous. Nominative case is associated with subjects and indefinite direct objects. Subjects of sentential complements are genitive. Dative, ablative, genitive, and instrumental can be non-canonical objects (Çetinoğlu and Butt, 2008), although their primary function is adjunct. In copular sentences, the nominal predicate, with or without an overt copular suffix, can bear any case marker except accusative.

Another morphological feature that parsing algorithms can benefit from is agreement. In Turkish, subjects and verbs must agree in number and person. There is an exception to this rule: a third person plural subject might agree with a verb in third person singular as well as a verb in third person plural.

To explore the question whether we can benefit from case and agreement features in parsing Turkish, we employ two different representation methods. First, we append case markers to nominal POS tags[3] to see if a more informative POS field could facilitate parsing (*Pos+Case*). Then, with the intuition that case markers alone could determine the function, we categorise nominals according to CASE instead of their POS (*Case*).

In the implementation, in order to represent case markers as categories we move them to the POS field. POS tags are moved to the morphological features field. For instance, In the CoNLL format[4], *çeki* 'cheque.Acc' has normally the representation in (5a). Appending CASE to POS results in (5b). When CASE replaces POS, the representation is as in (5c).

[3]These are namely nouns, proper nouns, pronouns, nominal participals, and infinitives.

[4]The columns are Form, Lemma, POS, Morphological Features respectively.

(3)

ID	Form	Lemma	POS	Morph. Feat.	Head	Dep. Rel.
1	açık	açık	Adj	_	2	MODIFIER
2	_	çek	Noun	A3sg|Pnon|Nom	3	DERIV
3	çekti	_	Verb	Zero|Past|A3sg	0	ROOT

(5)
a. çeki çek Noun A3sg|Pnon|Acc
b. çeki çek Noun|Acc A3sg|Pnon
c. çeki çek Acc A3sg|Pnon|Noun

This representation has two benefits. We can still use the POS tags as features for the parser, and after parsing, it is possible to restore the POS tags by switching them back. This allows us to evaluate our system against the standard gold data.

When combined with a joint parsing system, both approaches extend the use of the parser and practically carry it to a level between POS tagging and morphological analysis. We applied the CASE-POS replacement technique to agreement markers (*Agr*) hoping that the parser can learn and predict the relation between subjects and verbs better. We also collected the finite verbs under the `VFin` umbrella instead of `Verb` to distinguish verbs with an agreement marker from non-finite ones (*VFin*). We discuss the effects of those changes in Section 6.2.

5 Experimental Setup

5.1 Data Set

We use the METU-Sabancı Turkish Treebank (Oflazer et al., 2003) for training and ITU validation set (Eryiğit, 2007) for testing. The training and test sets consist of 5635 and 300 sentences respectively. There are no separate development sets. The original version of the treebank contains multiword expressions[5] where words that construct the multiword are attached together with an underscore. The POS and morphological features of a MWE are that of the last word of the MWE. Eryiğit et al. (2011) have created a detached version of the original treebank. In the detached version, multiword expressions are splitted into words, and POS and morphological features are assigned to the new words. They are dependent on the final word of the multiword with the relation MWE. (6a) and (6b) give the original and detached versions of *söz vermiştim* 'I have promised', respectively. Note that if a MWE consists derived words they will also be represented with multiple IGs. In our experiments we use the detached version of the treebank.

5.2 Tools

In order to parse data with predicted segmentation, POS and morphological features, the raw data is first passed through a morphological analyser (Oflazer, 1994) and then through a morphological disambiguator (Sak et al., 2008). Heuristic rules are used for some unknown types[6] and the rest of unknowns are considered to be nominative proper nouns. We adopt Bohnet's (2010) state-of-the-art graph-based parser as our *Pipeline* parser[7] and Bohnet and Nivre's (2012) transition-based parser as our *Joint* parser that can jointly handle POS tagging and dependency parsing.

5.3 Evaluation

The standard evaluation metrics labelled and unlabelled attachement scores (LAS and UAS) (Buchholz and Marsi, 2006) are not applicable to compare a predicted file to a gold file if the segment sizes are different. We handle this problem by using an evaluation tool based on IGs (Eryiğit et al., 2008). The unlabelled attachement score UAS_{IG} gives the ratio of IGs that are attached to the correct head, and the labelled attachement score LAS_{IG} gives the ratio of IGs attached to the correct head with the correct label. In cases where the morphology (segmentation, POS, and morphological features) of the head word is different from the gold one, an attachement is correct only if the dependent is attached to the correct word *and* the head IG has the gold main POS. Note that when gold segmentation and POS are used LAS_{IG} and UAS_{IG} are identical to the standard LAS an UAS respectively. We omit punctuation in evaluation.

[5] E.g., named entities, collocations, date-time expressions, noun-verb compounds as in (6).

[6] E.g. if a word ends with an apostrophe followed by the surface form of a case marker, the string before the apostrophe is the root of a proper noun and the case is determined from the surface form.

[7] We also ran baseline experiments with Bohnet's transition-based parser. The graph-based parser clearly outperforms it in the gold setting. When the parsers are provided with predicted POS tags and morphological features, the scores are comparable.

(6)

	ID	Form	Lemma	POS	Morph. Feat.	Head	Dep. Rel.
a.	4	söz_vermiştim	söz_ver	Verb	Pos|Narr|Past|A1sg	5	SENTENCE
b.	4	söz	söz	Noun	A3sg|Pnon|Nom	5	MWE
	5	vermiştim	ver	Verb	Pos|Narr|Past|A1sg	6	SENTENCE

6 Experiments and Analyses

We conduct 10-fold cross validation experiments on the training data and report the average scores for pipeline and joint parsers. Gold settings use gold segmentation, POS, and morphological features, whereas in predicted settings, all this information is predicted (either by the morphological analyser+disambiguator or by the joint parser). For systems we observe improvements on 10-fold cross validation experiments, we also give the test set results.[8]

6.1 Pipeline Experiments

In the first set of experiments, we examine the effect of using morphological features in parsing. Table 1 gives the average 10-fold cross validation scores on the training data. As discussed in Section 2, there are controversial results of using morphological features in parsing MRLs: although gold features help, predicted features might harm the accuracy. For Turkish, Eryiğit et al. (2008) have already shown that adding gold morphological features to Malt parser trained on the original treebank improves accuracy. Our findings are in line with theirs.

The first row of Table 1 gives the graph-based parser results when both the training and parsing data have morphological information. The predicted LAS_{IG} is 4.5% lower than the gold one. When the graph-based parser is trained on gold data with morphological features, but the features are not provided during parsing, there are 12.4% and 10.7% LAS_{IG} drops in the gold and predicted settings respectively. A drop in such a scenario is of course expected, but the impact of no morphology in parsing is huge as compared to many other MRLs (e.g., Seeker and Kuhn (2013) report 6.3%, 2.4%, and only 0.4% absolute drops in LAS for Hungarian, Czech, and German respectively). When the morphological information is not used in training at all, the parser can cope with the lack of morphological information better during parsing. Still, the gold and predicted LAS_{IG} scores are absolute 5-6% lower than a setting that uses morphology both in training and parsing.

	Gold		Predicted	
System	**LAS**$_{IG}$	**UAS**$_{IG}$	**LAS**$_{IG}$	**UAS**$_{IG}$
GB +T,+P	66.29	77.51	61.79	73.89
GB +T,-P	53.88	71.49	51.02	69.71
GB -T,-P	60.62	75.36	56.31	71.42

Table 1: The effect of using morphological features on the graph-based parser. Morphological features are used in neither training nor parsing (-T,-P), used in training but not provided in parsing (+T,-P), used both in training and parsing (+T,+P). Results given are the average 10-fold cross-validation scores on the training data.

6.2 Joint Parsing Experiments

Table 2 gives the training set 10-fold cross validation average scores for systems we experimented in this paper, as well as for previous work. It is observed that moving CASE to the POS field helps with a 0.3% absolute increase in the gold pipeline settings. Joint parsing results with gold features, are 1-1.5% absolute lower than the pipeline scores. This is expected; the gold setting for joint parsing is not exactly gold, as by definition the parser predicts POS tags during parsing instead of gold ones although the segmentation and morphological features are gold. As a result, they cannot beat purely gold settings.

If we have a closer look at the joint systems, we witness that only *Joint*$_{Case}$ outperforms *Joint*. *Joint*$_{Pos+Case}$ increases the tagset to be learned and predicted from 35 to 107 which is probably too fine-grained for the parser. Agreement markers, which are not directly related to grammatical functions like CASE, have a negative impact in the gold settings when used instead of `Verb`. Still, when agreement markers are used only to introduce an extra category, namely `VFin`, the scores come closer to the baseline of joint parsing with gold information, and even improves over the baseline LAS_{IG} in the predicted setting.

In the pipeline approach with predicted morphology, using CASE instead of nominal POS im-

[8] For replicability, experimental settings are available at `http://www.ims.uni-stuttgart.de/~ozlem/cetinogluDepling13.html`

	Gold		Predicted	
System	**LAS**$_{IG}$	**UAS**$_{IG}$	**LAS**$_{IG}$	**UAS**$_{IG}$
Pipeline	66.29	77.51	61.79	73.89
Pipeline$_{Case}$	**66.60**	**77.60**	62.07	74.00
Joint	64.61	75.83	62.21	73.86
Joint$_{Case}$	64.92	76.27	**62.58**	**74.35**
Joint$_{Pos+Case}$	63.99	75.45	62.02	73.76
Joint$_{Agr}$	63.65	74.95	61.32	73.17
Joint$_{VFin}$	64.44	75.68	62.34	72.59
Ery11-Ery12	65.90	76.00	58.3/61.1	70.70

Table 2: **Training set 10-fold cross validation average scores.** Gold scores Ery11 are taken from Eryiğit et al. (2011) and predicted scores Ery12 are taken from Eryiğit (2012). Ery12 (Eryiğit, 2012) gives an interval LAS$_{IG}$ corresponding 0% and 100% accuracy for MWE relations

	Gold		Predicted	
System	**LAS**$_{IG}$	**UAS**$_{IG}$	**LAS**$_{IG}$	**UAS**$_{IG}$
Pipeline	**68.92**	78.85	64.59	76.32
Pipeline$_{Case}$	68.86	**78.98**	65.00	76.35
Joint	66.14	76.86	63.77	75.06
Joint$_{Case}$	67.25	78.50	**65.19**	**77.05**
Ery11-Ery12	-	-	64.2/66.2	75.53

Table 3: **Testset scores.** Ery11 (Eryiğit et al., 2011) does not provide gold scores for testset. Ery12 (Eryiğit, 2012) gives an interval LAS$_{IG}$ corresponding 0% and 100% accuracy for MWE relations.

proves the labelled accuracy by 0.3% absolute for the training set. Letting the parser predict POS in the joint system adds 0.14 points more. The best score is achieved with *Joint$_{Case}$* which has a 0.3% absolute increase as compared to *Joint*. The difference between pipeline systems and joint systems are statistically significant both for LAS$_{IG}$ and UAS$_{IG}$, in the gold setting. When predicted data is used, *Pipeline$_{Case}$*, *Joint*, *Joint$_{Case}$* LAS$_{IG}$ scores are statistically significantly better than *Pipeline* ($p<0.05$, paired *t*-test).

The testset scores are given in Table 3. They follow the training set trend, except for the *Joint* system to our surprise. This is perhaps due to the different characteristics of test and training data. When we look at the breakdown of dependencies from 10-fold cross validation results in Section 6.3, we discuss a recall drop in some labels when they are parsed with the *Joint* parser. We do not look at the dependency distribution of the test data but if it is different from the training data then a possibly similar drop in the same labels might impact the overall score more. In parsing the test data with gold features, pipeline systems statistically significantly outperform joint systems. In the predicted setting, only *Joint* vs. *JointCase* UAS$_{IG}$ difference is statistically significant.

Both in Tables 2 and 3, predicted LAS$_{IG}$ scores from Eryiğit (2012) are given as an interval. In her experiments, the parser is trained on the original treebank (that is, no MWE relations are present in the training data) and tested on the detached version. She reports lower and upper bounds corresponding 0% and 100% accuracy for MWE relations. To compare our results to those of Eryiğit's, we also calculate the upper bounds with 100% MWE accuracy in our best performing system. When we accept all MWE labels correct[9] we achieve **64.49%** LAS$_{IG}$ on the average score of 10-fold cross validation on the training set and **66.46%** LAS$_{IG}$ on the testset for the *Joint$_{Case}$* system. For both the predicted and gold systems our parsers outperform previous work.

For comparability with other existing results, we also trained the *Pipeline* parser on the original version of the treebank which is used in the CoNLL 2007 Shared Task. Nivre et al. (2007) report **71.6%** LAS on the testset (excluding punctuation) for the best system (Titov and Henderson, 2007). Eryiğit (2012) increases the LAS to **71.98%** and the *Pipeline* parser outperform both systems with **72.53%** LAS .

6.3 Error Analysis

For a detailed error analysis we take into account the *Pipeline*, *Pipeline$_{Case}$*, *Joint*, and *Joint$_{Case}$* 10-fold cross validation results on the training set. In the predicted setting, scores from these four parsers are in ascending order (Table 2, predicted LAS$_{IG}$ column, first four rows). When we look at the dependency breakdown of pipeline and joint systems, we observe subjects and objects follow this trend, together with question particles, negative particles, and modifiers.

The dependencies that benefit from joint parsing the most are determiners. This is due to the fact that some frequently occuring determiners are ambiguous. For instance, *O* has the determiner ('that') and personal pronoun ('he/she/it') readings, and similarly *bu* 'this' is both a determiner and a demonstrative pronoun. Joint parsing lets the parser assign the correct POS to those words where the morphological disambiguator fails. Let-

[9] through a parameter in the evaluation script

Dependency	Precision	Recall
ABLATIVE.ADJUNCT	41.9	50.3
APPOSITION	48.3	15.0
CLASSIFIER	59.1	68.1
COORDINATION	53.0	48.4
DATIVE.ADJUNCT	40.5	45.8
DETERMINER	73.5	81.3
INSTRUMENTAL.ADJUNCT	24.6	21.0
INTENSIFIER	70.7	70.7
LOCATIVE.ADJUNCT	40.4	46.0
MODIFIER	60.3	58.3
MWE	63.5	58.1
NEGATIVE.PARTICLE	67.0	45.6
OBJECT	59.9	58.2
POSSESSOR	70.9	74.5
QUESTION.PARTICLE	71.5	62.8
SENTENCE	86.6	88.0
S.MODIFIER	49.4	46.1
SUBJECT	48.9	51.0
VOCATIVE	29.6	19.5

Table 4: The dependency breakdown of the 10-fold cross validation scores for *Joint*$_{Case}$ with predicted morphological information. Precision and recall are given in percent. Dependencies with less than 100 occurrences are omitted.

ting the parser predict CASE instead of POS causes some drop, but both precision and recall are still higher than both pipeline systems.

Another dependency with *Joint* as the most accurate system is coordination. CASE helps in *Pipeline*$_{Case}$ as compared to *Pipeline*, but causes an accuracy decrease when going from *Joint* to *Joint*$_{Case}$. The COORDINATION label attaches conjunctions to their conjunct to the right. The most frequent conjunctions comma and *ve* 'and' can be predicted with very high accuracy. When the *Joint* parser is used, there are slight improvements on attachements to head conjuncts with various POS tags and a systematic improvement on attaching conjunctions to head copulars and conditionals.

The precision of possessors does not change much with different systems, but the recall drops in *Joint*. That drop is recovered when *Joint*$_{Case}$ is applied. Intensifiers (e.g., particles *de* 'also, too', *bile* 'even') also have a similar trend. Precision, on the other hand increases with *Joint*.

A large subset of dependencies that suffers from the same drop is adjuncts. Dative, ablative, locative, and instrumental adjuncts commonly have drops in the *Joint* recall as compared to pipeline systems. Their precision, however, increases. When we look into the parser output, we see that the *Joint* system has systematically mistaken by assigning `Adj` to the `Verb` root of participles. Then all arguments attached to this incorrectly POS-tagged root are penalised by the evaluation script although most of the time attachements are correct.

The incorrect POS assignment problem disappears when the joint parser is trained on the CASE feature of nominals instead of their POS. This explains why the precision of adjuncts improves a bit more and their recall has a jump. The only exception is the precision drop in instrumental adjuncts. The reason could be nouns in instrumental case that behave as adverbs, such as *hızla* (`speed+Ins`, 'quickly'). The parser cannot learn to distinguish an instrumental adjunct from an adverbial modifier when `+Ins` is used as POS in *Joint*$_{Case}$.

The advantage of *Joint*$_{Case}$ over *Pipeline* is exemplified with a comparison in Figure 2. The *Pipeline* and *Joint*$_{Case}$ parse trees, together with POS tags and case markers are given in (a) and (b) respectively. The *Pipeline* parser relies on the morphological disambiguator output which incorrectly assigns the analyses (1b) to *kahveleri* and (1h) to *bende*. As a result, the parser assigns the incorrect labels to both dependencies.

On the other hand, the *Joint*$_{Case}$ parser replaces the case `Nom` with its prediction `Acc` in *kahveleri* and `Nom` replaces `Loc` in *bende*. These corrections result in predicting dependencies identical to gold ones. Note that the lemma of *bende* is still incorrect, but it does not affect the attachements.

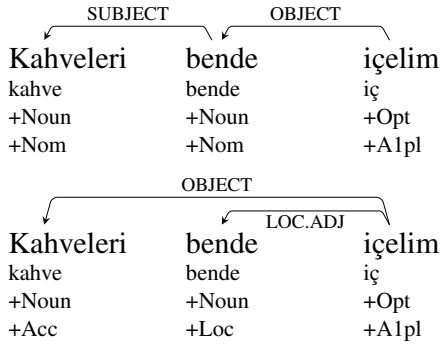

Figure 2: The (a) *Pipeline* and (b) *Joint*$_{Case}$ parse trees for the example sentence (1) *Kahveleri bende içelim* 'Let's drink coffee at my place.'

The dependency breakdown of the 10-fold cross validation parses for *Joint*$_{Case}$ with predicted morphological information is given in Table 4. In the

Turkish treebank representation, the root of a tree is the sentence-final punctuation. The main predicate of the sentence is attached to the sentence-final punctuation with the SENTENCE label. By far, this label is the easiest to predict with our systems. It is followed by determiners, intersifiers, possessors, and question particles, which are all local dependencies. Then come classifiers, coordination, modifiers, multiword expressions within a range of 50-65% precision and recall.

Despite getting improvements with the $Joint_{Case}$ system, grammatical functions are still quite low in accuracy. Except for objects, all such labels are below 50% precision and recall. This is due to both the free-word-order nature of Turkish and the ambiguous case-function mapping mentioned in Section 4.

And finally, appositions, vocatives, and instumental adjuncts are at the bottom of the accuracy ranking with scores going down to 20-30%. Their frequencies are also low and they have different POS and morphological features within the same class, which complicates parsers' learning.

7 Conclusion and Future Work

We have presented a set of experiments on parsing raw Turkish text. We argue the ideal method for parsing Turkish would be joint segmentation, POS tagging, morphological analysis, and dependency parsing. In this work we keep the segmentation fixed and first show using a joint POS tagging and parsing approach outperforms a pipeline approach in a realistic scenario. Then we come one step closer to the ideal case and attemp to incorporate some morphological features into joint prediction. As a second outcome, we show categorising nominals according to CASE instead of their POS improves parsing at all settings (gold vs. predicted, pipeline vs. joint). With the combination of joint parsing and CASE incorporation we not only show statistically significant improvements but also achieve the state-of-the-art parsing accuracy.

We believe these positive results prove there is room for improvement in predicting morphological features with a joint POS tagging and dependency parsing system. Even for the joint parsing experiments below the *Joint* baseline, more clever ways of integration into joint prediction might help achieve higher scores. Past research on MRLs present such cases. Bengoetxea and Gojenola (2010) show a simple integration of morphological features does not improve Basque parsing results on the first attemp, but taking advantage of the data representation and parser configuration changes the impact. Similarly, Tsarfaty and Sima'an (2010) has negative results initially for the impact of using agreement markers on Hebrew parsing. After they modify the way they use the morphological information, it actually helps.

In future work, we intend to explore ways to make more use of the joint parser and to apply the same or similar techniques to other MRLs such as German, Czech, and Hungarian.

We also want to add TedEval (Tsarfaty et al., 2012), which also supports mismatching system-gold segmentation, to our evaluation tools to verify our scores and to use a language-independent metric in a multilingual setting.

Acknowledgments

We thank Bernd Bohnet for his help on using the joint parser and Gülşen Eryiğit for providing us with the IG evaluation script. This work is funded by the Collaborative Research Centre (SFB 732) at the University of Stuttgart.

References

Bharat Ram Ambati, Samar Husain, Sambhav Jain, Dipti Misra Sharma, and Rajeev Sangal. 2010. Two methods to incorporate 'local morphosyntactic' features in Hindi dependency parsing. In *Proc. of the SPMRL Workshop of NAACL-HLT*, pages 22–30, Los Angeles, CA, USA.

Kepa Bengoetxea and Koldo Gojenola. 2010. Application of different techniques to dependency parsing of Basque. In *Proc. of the SPMRL Workshop of NAACL-HLT*, pages 31–39, Los Angeles, CA, USA.

Bernd Bohnet and Joakim Nivre. 2012. A transition-based system for joint part-of-speech tagging and labeled non-projective dependency parsing. In *Proc. of the EMNLP-CoNLL*, pages 1455–1465, Jeju, Korea.

Bernd Bohnet. 2010. Top accuracy and fast dependency parsing is not a contradiction. In *Proc. of COLING*, pages 89–97, Beijing, China.

Sabine Buchholz and Erwin Marsi. 2006. CoNLL-X shared task on multilingual dependency parsing. In *Proc. of CoNLL-X*, pages 149–164, Stroudsburg, PA, USA.

Özlem Çetinoğlu and Miriam Butt. 2008. Turkish non-canonical objects. In *Proc. of LFG08 Conference*, Sydney, Australia. CSLI Publications.

Jon Dehdari, Lamia Tounsi, and Josef van Genabith. 2011. Morphological features for parsing morphologically-rich languages: A case of Arabic. In *Proc. of the SPMRL Workshop of IWPT*, pages 12–21, Dublin, Ireland.

Gülşen Eryiğit. 2007. ITU validation set for METU-Sabancı Turkish treebank.

Gülşen Eryiğit. 2012. The impact of automatic morphological analysis & disambiguation on dependency parsing of Turkish. In *Proc. of LREC*, Istanbul, Turkey.

Gülşen Eryiğit, Joakim Nivre, and Kemal Oflazer. 2008. Dependency parsing of Turkish. *Computational Linguistics*, 34(3):357–389.

Gülşen Eryiğit, Tugay Ilbay, and Ozan Arkan Can. 2011. Multiword expressions in statistical dependency parsing. In *Proc. of the SPMRL Workshop of IWPT*, pages 45–55, Dublin, Ireland.

Yoav Goldberg and Michael Elhadad. 2009. Hebrew dependency parsing: Initial results. In *Proc. of IWPT*, pages 129–133, Paris, France.

Yoav Goldberg and Michael Elhadad. 2010. Easy-first dependency parsing of modern Hebrew. In *Proc. of the SPMRL Workshop of NAACL-HLT*, pages 103–107, Los Angeles, CA, USA.

Yoav Goldberg and Reut Tsarfaty. 2008. A single generative model for joint morphological segmentation and syntactic parsing. In *Proc. of ACL-HLT*, pages 371–379, Columbus, Ohio.

Spence Green and Christopher D. Manning. 2010. Better Arabic parsing: baselines, evaluations, and analysis. In *Proc. of COLING*, pages 394–402, Stroudsburg, PA, USA.

Jun Hatori, Takuya Matsuzaki, Yusuke Miyao, and Jun'ichi Tsujii. 2011. Incremental joint POS tagging and dependency parsing in Chinese. In *Proc. of IJCNLP*, pages 1216–1224, Chiang Mai, Thailand.

John Lee, Jason Naradowsky, and David A. Smith. 2011. A discriminative model for joint morphological disambiguation and dependency parsing. In *Proc. of ACL-HLT*, Portland, Oregon, USA.

Zhenghua Li, Min Zhang, Wanxiang Che, Ting Liu, Wenliang Chen, and Haizhou Li. 2011. Joint models for Chinese POS tagging and dependency parsing. In *Proc. of EMNLP*, pages 1180–1191, Edinburgh, Scotland, UK.

Yuval Marton, Nizar Habash, and Owen Rambow. 2010. Improving Arabic dependency parsing with lexical and inflectional morphological features. In *Proc. of the SPMRL Workshop of NAACL-HLT*, pages 13–21, Los Angeles, CA, USA.

Joakim Nivre, Johan Hall, Sandra Kübler, Ryan Mac Donald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret. 2007. The conll 2007 shared task on dependency parsing. In *Proc. of the CoNLL Shared Task Session of EMNLP-CoNLL*.

Kemal Oflazer, Bilge Say, Dilek Zeynep Hakkani-Tür, and Gökhan Tür. 2003. Building a Turkish treebank. In Anne Abeille, editor, *Building and Exploiting Syntactically-annotated Corpora*. Kluwer Academic Publishers, Dordrecht.

Kemal Oflazer. 1994. Two-level description of Turkish morphology. *Literary and Linguistic Computing*, 9(2):137–148.

Haşim Sak, Tunga Güngör, and Murat Saraçlar. 2008. Turkish language resources: Morphological parser, morphological disambiguator and web corpus. In *Proc. of GoTAL 2008*, pages 417–427.

Wolfgang Seeker and Jonas Kuhn. 2011. On the role of explicit morphological feature representation in syntactic dependency parsing for German. In *Proc. of IWPT*, pages 58–62, Dublin, Ireland.

Wolfgang Seeker and Jonas Kuhn. 2013. Morphological and syntactic case in statistical dependency parsing. *Computational Linguistics*, 39:23–55.

Sebastian Sulger, Miriam Butt, Tracy Holloway King, Paul Meurer, Tibor Laczkó, György Rákosi, Cheikh Bamba Dione, Helge Dyvik, Victoria Rosén, Koenraad De Smedt, Agnieszka Patejuk, Özlem Çetinoğlu, I Wayan Arka, and Meladel Mistica. 2013. Pargrambank: The pargram parallel treebank. In *Proc. of ACL*, Sofia, Bulgaria.

Ivan Titov and James Henderson. 2007. Fast and robust multilingual dependency parsing with a generative latent variable model. In *Proc. of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 947–951.

Reut Tsarfaty and Khalil Sima'an. 2008. Relational-realizational parsing. In *Proc. of COLING*, pages 889–896, Manchester, UK.

Reut Tsarfaty and Khalil Sima'an. 2010. Modeling morphosyntactic agreement in constituency-based parsing of modern Hebrew. In *Proc. of the SPMRL Workshop of NAACL-HLT*, pages 40–48, Los Angeles, CA, USA.

Reut Tsarfaty, Djamé Seddah, Yoav Goldberg, Sandra Kuebler, Yannick Versley, Marie Candito, Jennifer Foster, Ines Rehbein, and Lamia Tounsi. 2010. Statistical parsing of morphologically rich languages (SPMRL) what, how and whither. In *Proc. of the SPMRL Workshop of NAACL-HLT*, pages 1–12, Los Angeles, CA, USA.

Reut Tsarfaty, Joakim Nivre, and Evelina Andersson. 2012. Joint evaluation for morphological segmentation and syntactic parsing. In *Proc. of ACL*, Jeju, Korea.