

Neural Disambiguation of Lemma and Part of Speech in Morphologically Rich Languages

José María Hoya Quecedo, Maximilian W. Koppatz, Roman Yangarber

University of Helsinki

Department of Computer Science

josemhoya@gmail.com, max.koppatz@gmail.com, roman.yangarber@helsinki.fi

Abstract

We consider the problem of disambiguating the lemma and part of speech of ambiguous words in morphologically rich languages. We propose a method for disambiguating ambiguous words in context, using a large un-annotated corpus of text, and a morphological analyser—with no manual disambiguation or data annotation. We assume that the morphological analyser produces multiple analyses for ambiguous words. The idea is to train recurrent neural networks on the output that the morphological analyser produces for *unambiguous* words. We present performance on POS and lemma disambiguation that reaches or surpasses the state of the art—including supervised models—using *no manually annotated data*. We evaluate the method on several morphologically rich languages.

Keywords: Statistical and Machine Learning methods, Part-of-Speech Tagging, Morphology

1. Introduction

The problem of disambiguation is defined as selecting the correct analysis from a set of possible analyses for a word in a sentence—e.g., from among the analyses produced by a morphological analyser. Disambiguation is performed by utilizing information in the surrounding context.

Morphological analysers are commonly used in various NLP applications. These normally produce a significant amount of ambiguous analyses. In this work we tackle the problem of disambiguation by training a model for predicting the correct part-of-speech (POS) and lemma. We show that for the majority of cases, this is sufficient to disambiguate from the set of possible analyses.

We use manually annotated data only for *evaluation*, which means that to train our model we need only a morphological analyser for the language and an unlabelled corpus.

The main idea of our approach is to use bidirectional LSTMs (Gers et al., 2000)—BiLSTMs—to disambiguate the output of morphological analysers, by utilizing only the unambiguous outputs during the training procedure. We train bidirectional models using a sequence of embeddings for the surface form for each target word. The objective of the network is to produce output probability distributions over the possible POS tags and lemmas. The model is trained using only the unambiguous input tokens; the loss is computed only for those unambiguous instances. Ambiguous tokens are not considered as target tokens during training.

Since we only input unlabelled data for training, the quality of the model itself is only affected by the amount of available unlabelled data for the language. In our experiments, we evaluate our models on manually annotated data sets for Finnish, Russian and Spanish. For Finnish and Russian, at least, annotated (i.e., disambiguated) data is in limited supply, whereas for all three languages unlabelled data is in abundant supply.

The paper is organized as follows. In Section 2. we point to some relevant prior work. In Section 3. we describe the problem of morphological ambiguity and provide a brief

motivation for the interest in the problem. In Section 4. we provide a classification for the different types of ambiguity that appear in the corpus, as well as an analysis of the viable and appropriate strategies for each type of ambiguity. Section 5. describes our data pre-processing steps and model architecture. Section 6. specifies our experimental setup, as well as the parameters used in training. In Section 7. we discuss the results obtained from the experiments. Section 8. concludes with current directions of research.

2. Related work

There is an abundance of work on disambiguation in the context of various NLP tasks, we focus on just a few relevant ones here.

The work of Yatbaz and Yuret (2009) is conceptually similar to ours. Their work presents a probabilistic model for selecting the correct analysis from a set of morphological analyses for Turkish. Turkish and Finnish, as synthetic agglutinative languages, share the problem of a high number of possible analyses for a given word. This limits the amount of unambiguous data and presents a bigger problem than analytic or morphologically poor synthetic languages such as English. The LSTM based approach by Zalmout and Habash (2017), for Arabic, is also similar to our method. They train a POS tagging model on an annotated corpus, using added features, and use the resulting model to disambiguate a morphological analyser, achieving a lemma accuracy of 96.8%. The POS tagger by Inoue et al. (2017) for Arabic utilizes a form of multi-task learning. Tkachenko and Sirts (2018) present another neural morphological tagger, for Estonian, in which the output of an analyser is also used to augment the input to their neural models.

In contrast to the above mentioned neural models, we use the unambiguous outputs of the analyser to learn to disambiguate remaining ones, instead of learning a POS tagger on an annotated corpus.

3. Problem description

3.1. Definitions

Throughout this work, we make use of the following concepts:

- The *part-of-speech* (POS) of a word (of a surface form) is its morpho-syntactic category or class. This indicates the role the word plays in the sentence, as well as the inflectional paradigm—the pattern of inflection—that the word follows. Examples of POS are: noun, verb, and adjective.¹
- The *lemma* is the canonical, or “dictionary,” form of a word. For example, for nouns the lemma is the nominative case singular and for verbs the lemma is the infinitive.²
- A *surface form* is the form in which the word appears in text. The surface form may be an inflected form of the lemma, or may be identical to the lemma; for uninflected POSs, the surface form is always identical to the lemma.
- *Morphological tags* are values that the morphological analyser assigns to morphological features of the word. For example, the feature *number* may have values such as *singular* and *plural*; the feature *case* may have values such as *nominative* and *genitive*, depending on the feature inventory of the language.

Morphological analysis is the task of breaking down a surface form into its lemma, POS and morphological features (tags), by means of a morphological analyser. As an example, consider the Finnish surface form “*kotiin*” (into/toward home). A morphological analysis of “*kotiin*” would be:

`koti+N+Sg+Ill`

This indicates that the lemma is `koti`, the POS is `N` (Noun), and the morphological features are `Sg` (singular number) and `Ill` (illative case, meaning “into/toward”).

3.2. Ambiguity

Natural language is inherently ambiguous, and there are many ways in which ambiguity manifests itself. For written text, we have several types of ambiguity. *POS* ambiguity is a kind of syntactic ambiguity, where a word may be considered to have one of several syntactic roles inside a sentence. *Lemma* ambiguity occurs when a surface form is a form of more than one lemma. *Morphological* ambiguity occurs when a surface form has several possible analyses—several sets of morphological tags. *Word sense* ambiguity—when a single lemma may have several different meanings.

In spoken language, other kinds of ambiguities exist, such as homophones—two words which are written differently but are pronounced the same. Spoken language ambiguity

¹Many languages, including those we work with, distinguish *open* vs. *closed* POS classes. In morphologically rich languages, open POSs are heavily inflected, whereas closed classes are not inflected, or have very limited inflection.

²This is dependent on the language—the former holds true for Finnish and Russian, which are the languages with which we experiment in this paper, but not for other languages such as Latin.

is outside the scope of our work, we concentrate on written text.

One example of ambiguity is the Finnish surface form “*tuli*”, which has the following analyses:

`tuli (fire)` Noun, nominative, sing. ||

`tulla (come)` Verb, indicative, active, past, 3rd person, sing.

This exhibits all of the above kinds of ambiguity: POS, lemma, morphological tags, and word sense are all ambiguous.

Disambiguation is a central problem in many NLP tasks, for many reasons. In particular, morphological disambiguation in morphologically rich languages is crucial for translation. In our application,³ where we build tools to aid in language learning, when a student points at an unfamiliar surface form in the text which happens to be ambiguous, we need to identify the lemma *appropriate to the context*—so as not to confuse the learner with extraneous translations.⁴ Especially in agglutinative, morphologically rich languages such as Finnish, unambiguous lemmatization is central for NLP applications that build a vocabulary from corpora. For these languages the size of the vocabulary becomes very large without lemmatization. If the lemmatization is ambiguous, then subsequent models are based on an inaccurate vocabulary.

Our approach is based on the assumption that the context *primes* the selection of the appropriate reading from a set of several readings for an ambiguous surface form. By “*priming*,” we mean the following: a simple experiment with Google’s translator shows that the ambiguous word `белки`, is easily disambiguated by its immediate context. The surface form has two lemmas: “`белка`” (squirrel) and “`белок`” (protein). Google easily translates “`белки и медведи`” as “squirrel and bears”, whereas it translates “`белки и углеводы`” into “proteins and carbons.”

Thus, Google’s translation problem subsumes the disambiguation problem that we are trying to solve; in fact, Google’s translator could be viewed as a “poor man’s solution” to the disambiguation problem. However, because we are trying to solve the simpler problem—disambiguation—key point is that we may be able to solve it with a more lightweight solution. This would offer 3 benefits: A. we could achieve it with fewer and cheaper resources—translation requires supervision from massive parallel corpora; B. we may be able to achieve it with simpler models; and C. we may be able to achieve better performance on disambiguation, than if we tried to use a full translation machine to perform disambiguation. We further rely on the assumption that a large corpus will contain enough unambiguous contexts for each POS and for each lemma, so that the model should be able to learn to disambiguate the ambiguous instances.

4. Types of ambiguity

We discuss briefly a taxonomy of the types of ambiguity that are of interest to us. Additional examples are given

³`revita.cs.helsinki.fi`

⁴To put it in other words: in our work, we are not concerned with word-sense ambiguity *alone*—only in conjunction with POS ambiguity or lemma ambiguity.

	Declinable-Declinable		Declinable-Indeclinable		Indeclinable-Indeclinable	
≠ POS = lemma	POS amb.	dis-	POS amb.	dis-	POS amb.	dis-
= POS ≠ lemma	lemma amb.	dis-	n/a		n/a	
≠ POS ≠ lemma	either		POS amb.	dis-	n/a	
= POS = lemma	neither		n/a		n/a	

Table 1: Viable approaches for each type of ambiguity.

in Appendix A, for several languages. In many cases, the problem of disambiguation can be reduced to one of two problems: POS tagging or lemmatization—given a surface form in context (running text), find its POS or lemma, respectively. We outline the main types of morphological ambiguity, and whether we can use one approach or the other to resolve it.

We classify lemmas into two types—depending on whether they accept inflectional morphemes: *declinable* lemmas accept them, and *indeclinable* lemmas do not. Thus, an indeclinable lemma has only one surface form. Declinable lemmas can have many surface forms.

We use the term *reading* to denote a unique combination of lemma and POS.

We divide surface form ambiguities into three categories in the following subsections: two (or more) declinable lemmas, one declinable and one indeclinable lemma, or two indeclinable lemmas.

4.1. Surface forms with two declinable lemmas

This is the easiest case to train for, since, in general, the sets of surface forms derived from the two lemmas rarely overlap. For example, Finnish surface form FI “tuli” has two readings, as above:

tuli (*fire*) Noun, nominative, sing. ||
tulla (*come*) Verb, indicative, active, past, 3rd person, sing.

In this example, the lemmas and the POS’s of the readings are both different. This is the most common type of ambiguity, and either method (POS or lemma disambiguation) can be applied.

If the lemmas of the readings are identical, we cannot use lemmatization to resolve the ambiguity, and must resort to POS disambiguation, e.g.:

RU “ЗНАТЬ”: (*know*) Verb || (*nobility*) Noun

Conversely, if the POS’s of the readings are the same, but the lemmas are different, we cannot use POS tagging to resolve the ambiguity:

When two readings are the same for a surface form—i.e., the lemma and POS are the same, but the morphological tags are different—our methods are not suitable to disambiguate: e.g., FI “nostaa”:

nostaa Verb, infinitive ||
nostaa Verb, present, indic., 3rd, sing.

Lastly, we turn to word-sense ambiguity. For example, in English, the word/lemma “spirit” may mean “soul” or “alcohol”. These are unrelated semantically, but have the same lemma, same POS, and follow identical inflectional patterns. Although this type of ambiguity is also important for translation, it is outside the scope of this paper.⁵

To sum up, we are concerned with disambiguating among different *readings*—i.e., POS or lemma disambiguation. We are not concerned with disambiguating different possible morphological tags of a given reading, nor with disambiguating multiple word senses of a given reading.

4.2. Surface forms with one declinable and one indeclinable lemma

In this case, trying to predict the lemma may be less effective: although the lemmas may be different, *every* instance of the reading with the indeclinable lemma is ambiguous—since it always “drags along” the other readings with it.

Finnish and Russian have many instances of such surface forms. In Finnish, many adverbs or post-positions originate historically from an inflected form of a semantically related noun. For example, FI “jälkeen”:⁶

jälkeen (*after*) Post-position ||
jälki (*into a footprint*) Noun, illative, sing.

Thus, every occurrence of the post-position “jälkeen” drags along with it the readings for the illative case of “jälki” (which is also a valid reading of “jälkeen”).

However, the model can still hope to learn that the POS of this surface form is post-position, since other *unambiguous* post-positions may occur in similar contexts elsewhere in the corpus. Thus, POS tagging is an effective solution to this type of ambiguity.

The POS determines whether the reading is declinable or indeclinable. Thus a surface form cannot have a declinable and an indeclinable reading with the same POS, as seen in Table 2, and in fact such instances do not appear in the corpus.

4.3. Two indeclinable lemmas

If the readings are both indeclinable—neither can be inflected—and if their lemmas are different, there is no ambiguity, as the surface forms will always differ. If the lemma and POS are the same, the readings must be trivially identical, since there are no other morphological features. If the POS is different, we can disambiguate via POS tagging, as in the previous category, 4.2.. It is important to note that, since these readings *always* go together, in our work we consider these ambiguities to be “uninteresting”, since they can be considered extra-morphological.⁷

⁵However, we are interested in distinguishing the Noun “spirit” from the Verb “spirit”, which has a different meaning—“to move away briskly or secretly”—but also crucially has a different POS.

⁶The post-position “jälkeen” (after) is an ossified form of an inflection of the noun “jälki” (footprint)—in the sense of “after” meaning “in the footsteps of”. However, most analysers will analyse the post-position “jälkeen” as a separate lemma, not explicitly linked to its old nominal origin.

⁷As is the case for English “around,” which can be either an adverb or a preposition.

	Declinable- Declinable	Declinable- Indeclinable	Indeclinable- Indeclinable
≠ POS = lemma	8.78%	1.63%	6.47%
= POS ≠ lemma	8.93%	0.00%	0.00%
≠ POS ≠ lemma	40.29%	27.88%	0.00%
= POS = lemma	6.04%	0.00%	0.00%

Table 2: Incidence of each type of ambiguity in the Finnish corpus.

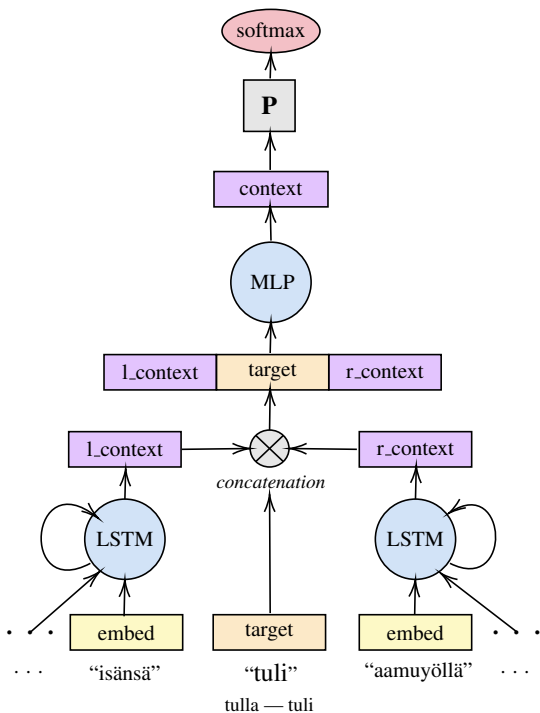


Figure 1: Overview of the network (see Section 5.2.).

Table 1 provides a summary of the effective approaches for each category.

In table 2, we see that POS tagging can effectively disambiguate most cases, except those in which the POS is the same across readings.

5. Model

We next turn to the technical description of our approach. First, we outline the steps involved in preparing the data for our model. We then proceed to present the architecture of our model, and the training procedure.

5.1. Data pre-processing

First, we tokenize each document as a flat list of surface forms (tokens).

We then use morphological analysers to obtain the readings of each surface form. For Finnish, we use analysers

from the Giellatekno platform (Moshagen et al., 2013)⁸. Giellatekno analysers are based on Two-level Morphology, by Koskenniemi (1983). For Russian, we use the analyser from Klyshinsky et al. (2011). For Spanish, we use the analyser from Forcada et al. (2011).

Since the goal is to *disambiguate* the output of the analyser, the coverage of said analyser—the percentage of tokens that have an analysis—is a relevant concern. The Finnish, Russian and Spanish analysers have 95.14%, 97.79% and 96.78% coverage, respectively. Most of the unknown tokens are foreign or misspelled words.

For Finnish—which has compounding—we split the surface form of the compounds into their “maximal” pieces, i.e., the largest parts for which there is a lemma in the analyser’s lexicon. For example, the Finnish compound word *eläinlääkäriasema* (“veterinary clinic”) is made up of three elementary stems: *eläin* (“animal”) + *lääkäri* (“doctor”) + *asema* (“station”). However, since the analyser has *eläinlääkäri* (“veterinarian”) in its lexicon, we split as *eläinlääkäri* + *asema*. This helps us keep the vocabulary smaller—since there is a potentially infinite number of possible compounds in Finnish—while keeping the meaning of commonly used compounds, which usually differs a little from that of the sum of its parts.

For Russian, this is not a concern, as there are generally no compound words. There may be cases in which a lemma is formed by joining two other lemmas, but this is considered a new lemma in its own right. The same applies to Spanish, where we additionally have clitic pronouns attached to verbs and prepositional contractions (preposition + article), which are treated as separate tokens.

While we do preserve information about sentence boundaries in the form of punctuation, we do not explicitly preserve sentence structure in terms of the training window. We found that several sentences in our corpora were too short to provide the contextual information necessary for disambiguating the target words, and that this information was partially found in the adjacent sentences. Instead, we make a sliding window of radius r over this list of tokens, i.e. we take r tokens to the left and r tokens to the right of some given target token, as well as the token itself.

Tokens are selected as targets for the training set only if they are unambiguous. Each training instance consists of said window, and the label for the target word, given by the analyser—the lemma or POS, depending on the desired target for the model. The target for the lemma is the index of said lemma in our vocabulary.

For the test set, we instead select only the ambiguous tokens, since the unambiguous ones will trivially give us a 100% accuracy. Each testing instance consists of the window, the possible labels and the true label for the target word.

We then obtain the word embeddings for each surface form in the window using the FastText (Bojanowski et al., 2017) Common Crawl⁹ pre-trained models for each language. This allows us to get an embedding even for out-of-vocabulary words, and to efficiently get embeddings for

⁸giellatekno.no

⁹https://commoncrawl.org/

Window size	21
Embedding size	300
LSTM hidden units	512
LSTM layers	1
MLP hidden neurons	1024

Table 3: Settings for the network.

Batch size	50
Dropout rate	0.1
Adam β_1	0.9
Adam β_2	0.999
Adam α	0.0001
Adam ϵ	0.001
Epochs	20

Table 4: Settings for the training hyper-parameters.

Finnish, which has a very large surface form vocabulary in our corpus—around 2 million unique tokens. FastText allows us to avoid using an embedding matrix and instead obtain the embeddings dynamically during training.

For the positions in the window which go beyond the limits of the document, we insert a zero-valued embedding as padding.

Each language has its own morphological analyser, and their outputs differ slightly. For compatibility between different analysers and our model, we map all POS to a common universal set. This also allows us to simplify the problem, by aggregating POS that fulfil a similar role, such as: postpositions + prepositions \rightarrow adpositions.

We use a set of 10 POS: noun, pronoun, numeral, adjective, verb, adverb, adposition, conjunction, punctuation, other (a catch-all category for things like acronyms or symbols).

Lemmas which are composed entirely of numerical digits are assigned a special embedding for numbers, since we consider them to always have a more or less equivalent role in the context.

Ambiguities between common nouns and proper nouns are ignored, as names are out of the scope of what we try to accomplish here and should be solved using Named Entity Recognition (NER) techniques instead.

We keep punctuation in order to recover the sentence boundaries within the window, as these may still prove useful to generate a proper context.

5.2. Architecture & training

Our model architecture is based on context2vec (Melamud et al., 2016), which itself is a modification of the original word2vec CBOW model (Mikolov et al., 2013). In context2vec, the context for each word is computed using a bidirectional LSTM, rather than as a vector average (as in word2vec), which enables the embeddings to capture sentence structure and word order, rather than only word co-occurrence.

The training procedure is analogous to that of word2vec, since our objective is similar: given a context, compute the probability that the word belongs to that context, for each word in the vocabulary—in our case, instead of word

tokens, we compute the probability for the lemma or the POS. Thus, our input and target vocabularies are different, in contrast to word2vec.

The architecture borrows some ideas from neural machine translation (NMT) encoder-decoder models, such as that developed by Google (Wu et al., 2016). In that model, encoding the context of a token into one vector is enough to be able to translate—and therefore disambiguate—that token. We therefore use the encoder part of the architecture to capture the necessary information to disambiguate a token. The model consists of three trainable parts:

- Bi-LSTM which produces the left and right context embeddings.
- multi-layer Perceptron (MLP), which merges these into a single context embedding.
- projection matrix, to transform the context embedding into scores for all possible labels.

Each training instance consists of a window of surface form embeddings around an unambiguous target word, and the label for such word, defined as the index of the corresponding lemma or POS in the vocabulary.

To obtain the predictions, we proceed as follows. First, we feed the window from the beginning to the target word to the left LSTM, and from the end to the target word to the right LSTM. Their hidden states serve as the left and right context embeddings.

We then concatenate the left and right context with the surface form embedding for the target word, and feed the result to the MLP. This “residual” connection, where some input is fed to several layers of the network, is also a concept from NMT, and is done in order to separate important parts of the input from the encoded state.

Next, the output of the MLP is multiplied by the projection matrix, to get an array of scores of length equal to the number of possible labels.

Finally, we apply a softmax function to get the probability distribution for all labels. An overview of the model can be observed in Figure 1.¹⁰

To obtain the loss for the model, we compute the cross-entropy between the predicted probability distribution and the real distribution, which is the one-hot encoding of the true label index.

6. Experiments

6.1. Data

The data we use are obtained from two different sources. The annotated evaluation data is acquired from the Universal Dependencies Treebank, (Nivre et al., 2018). These data are in the CoNLL-U 2006/2007 format, (Nivre et al., 2007). The annotations in the data are used purely for the evaluation of the resulting models. For Finnish, as the annotated data sets were quite small, we used a collection of 600K

¹⁰The input (bottom line) is a window of words in Finnish: “. . . isänsä tuli aamuyöllä . . .” (“. . . his father arrived in the early hours . . .”) The *target* surface form, for which we will try to get a prediction, is ambiguous: it may be a verb (arrive) with lemma “tulla”, or a noun (fire) with lemma “tuli.” (This is the same as the example discussed in Section 3.2..)

Language	Target	Ambiguity	Precision	Recall	F_1 score
Finnish	POS	8.9	0.81	0.79	0.79
	Lemma	8.8	0.80	0.75	0.76
Russian	POS	7.7	0.85	0.84	0.84
	Lemma	10.2	0.82	0.77	0.80
Spanish	POS	15.3	0.83	0.81	0.81
	Lemma	15.5	0.82	0.75	0.75

Table 5: Evaluation metrics for each model. The column *Ambiguity* indicates the percentage of ambiguous tokens in each corpus.

Language	Target	Blind	Guided	Token	SOTA
Finnish	POS	73.7	79.1	98.1	97.7 (2018)
	Lemma	40.4	76.5	97.9	95.3 (2018)
Russian	POS	81.9	83.9	98.6	96.9 (2016)
	Lemma	60.7	83.4	98.3	81.7 (2015)
Spanish	POS	74.9	80.7	97.0	89.0 (2015)
	Lemma	55.2	74.7	96.1	88.0 (2015)

Table 6: The columns mean: *Blind*: percentage of ambiguities resolved with “blind” predictions—without using the analyser output; *Guided*: percentage of disambiguated by picking the top prediction from the analyser output; *Token*: overall *token-level* accuracy, by applying the best method. *SOTA*: for comparison, shows the state-of-the-art results.

proprietary news articles for training the model, after processing the text with the Finnish morphological analyser. The Russian annotated data set was large enough to use its predefined train-test split. Technically this split would not have been necessary, as we do not use annotations from data as labels.

6.2. Experimental setup

For each language, we trained two separate models: one to predict the correct POS, and one to predict the lemma. We learn the model by training on the unambiguously analysed tokens; we do not train on the ambiguous instances—this allows us to explore the unsupervised approach, with no need for manual disambiguation.

In each case, we evaluate our models by two metrics. Firstly, we pick the analysis with the highest value in the softmax output probability vector. We call this the “blind” disambiguation. Secondly, rather than picking the highest-scoring softmax output overall, we rather pick the highest-scoring output from the same output probability vector, but only from among the options deemed possible by the morphological analyser. For example, if the model is predicting the POS, the blind approach selects the POS that receives the highest output score from all possible POSs in the language. The analyser-based approach selects the highest scoring POS only from those POS values that are among the possibilities admitted by the morphological analyser for the given surface form. We proceed analogously for the lemma-based models.

The “blind” predictions are thus equivalent to plain POS tagging and lemmatization.

We evaluate each model with the manually annotated, disambiguated corpus for each language. We compute both evaluation metrics (precision, recall and F_1 score) as well

as the percentage of correct predictions, for direct comparison with the state of the art.

In addition, we evaluate each metric with respect to a “confidence” measure, defined as the probability given by the softmax function for each prediction. To this end, we set a confidence threshold θ_{conf} such that any prediction with confidence below that threshold will be deemed invalid. In doing so, we wish to test whether the more confident predictions will have a higher precision without a significant loss in recall, for applications where the goal is to obtain the highest possible precision.

Table 3 details the parameters used for the network. Increasing the number of trainable parameters yields no significant increase in accuracy. It is possible that with a more complex model the prediction accuracy could be slightly higher.

Table 4 details the training hyper-parameters. We use the Adam optimizer, (Kingma and Ba, 2015), to minimize the loss.

7. Results

For the three languages on which we performed an evaluation of our models, we significantly reduced the number of remaining ambiguities. Table 5 illustrates the results of our experiments in terms of number of ambiguities and evaluation metrics. Table 6 shows a comparison between our results and the state of the art.

While the Finnish and Russian analysers are much less ambiguous than the Spanish one, our model is able to disambiguate the Spanish output to very nearly the same token-level accuracy. Thus, our method is not reliant on a low percentage of ambiguity to begin with, but instead other factors—such as the overlap in surface forms for a given pair of lemmas—are much more relevant.

Language	Noun	Adjective	Verb	Adverb	Other
Finnish	75.8	82.8	78.3	74.7	75.5
Russian	77.4	79.8	90.7	82.6	92.5
Spanish	82.6	71.6	86.3	86.6	82.7

Table 7: Accuracy (percent) for each POS.

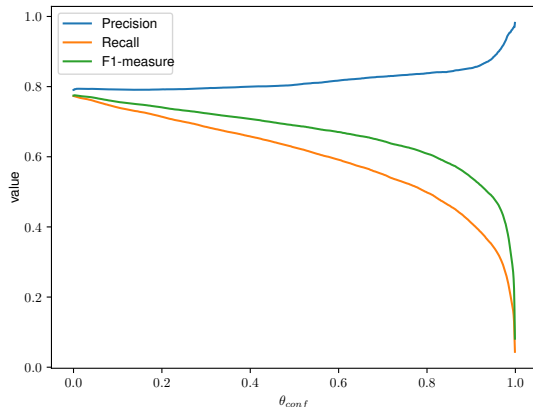


Figure 2: Finnish POS confidence vs. metrics.

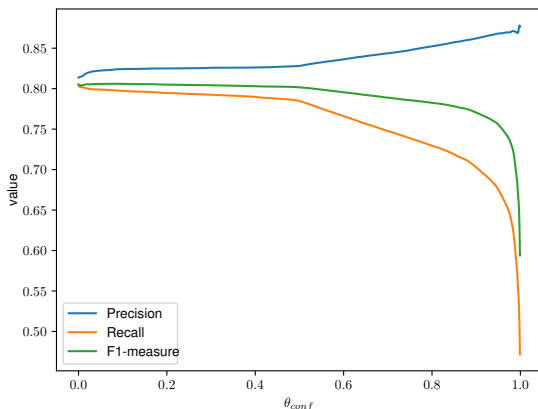


Figure 3: Russian POS confidence vs. metrics.

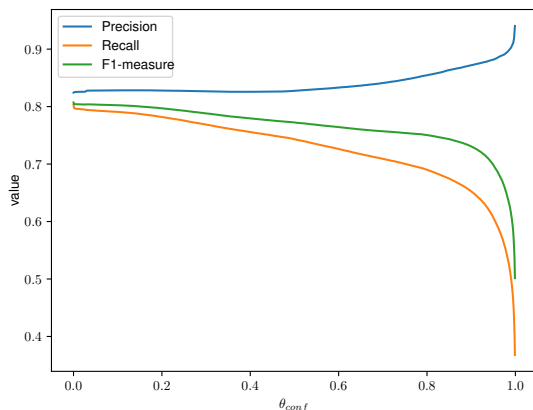


Figure 4: Spanish POS confidence vs. metrics.

For Russian, the best result to date for POS tagging was reported by Dereza et al. (2016), achieved using Tree-Tagger, (Schmid, 1994), at 96.94%. We could not find lemmatization results for Russian, but the work by Korobov (2015) solves the broader problem of morphological ambiguity with an accuracy of 81.7%.

For Finnish POS tagging and lemmatization, the TurkuNLP neural model (Kanerva et al., 2018) achieves 97.7% and 95.3% accuracy, respectively, evaluated on the same dataset as our method.

For Spanish POS tagging and lemmatization, the model by Carreras et al. (2004) achieves an accuracy of 89% and 88%, respectively, according to the evaluation done by Parra Escartín and Martínez Alonso (2015).

As for the confidence analysis, we see that, for every language, we can in fact build a POS model which has very high precision (>0.9) at the cost of being unable to obtain a prediction for a fraction of the instances. Figure 2, Figure 3 and Figure 4 show the results for Finnish, Russian and Spanish, respectively.

8. Conclusions

We have shown that the output of morphological analysers can be disambiguated to a significant degree for Finnish, Russian and Spanish. The requirements for this procedure are: the language must have a morphological analyser, there must exist a text corpus, and preferably a small amount of annotated data for evaluation purposes. The same procedure we used should perform comparatively for any language with a morphological analyser, assuming it is of sufficient quality—unknown tokens must rely on the less accurate “blind” predictions for inference. There are many morphologically rich languages that could benefit from this, such as other Uralic languages, Turkic languages, many Indo-European languages, etc. There is limited annotated training data for many of these languages, but morphological analysers are available for most of them.

The quality of the analyser in terms of percentage of unambiguous output does affect the final total token accuracy. The difference between the two cases end result presented in this work was small in the end. It is unclear how much ambiguity will begin to significantly impair our method.

Named Entity Recognition (NER) could theoretically be used in conjunction with our procedure to further disambiguate the proper noun analyses.

We have achieved different performance depending on whether the objective used was disambiguating the lemma or POS. We have seen that different types of ambiguity are solved to varying degrees by predicting either POS or lemma. A natural next step would be to combine the two different models in an ensemble model.

In table 2 we saw that, although POS tagging works for most of the cases, around 9% of the ambiguities are only solvable by lemma prediction. Since it is possible to identify these instances during inference, an ensemble solution could use the lemma prediction model to disambiguate these.

Moreover, around 6% of the instances currently cannot be disambiguated using either method.

This puts the upper limit on accuracy to 85% for the better model (POS prediction). Using an ensemble model to also capture the lemma-only ambiguities would therefore push this limit to 94%.

Another approach we have explored is the use of multi-task learning to predict both POS and lemma at the same time. We tried a naïve approach, reusing the LSTM parameters and alternating between the two different objectives during training. So far this has been somewhat unsuccessful, yielding an accuracy around 10% lower than that of either of the single-task models, but we believe there is still much room for improvement.

To push the performance nearer to 100%, it will be necessary to make a model that predicts morphological tags, either as an addition to the existing models, or as a standalone model that we can then invoke for these instances where the POS and lemma are the same.

9. Bibliographical References

- Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.
- Carreras, X., Chao, I., Padró, L., and Padró, M. (2004). Freeling: An open-source suite of language analyzers. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC'04)*.
- Dereza, O., Kayutenko, D., and Fenogenova, A. (2016). Automatic morphological analysis for Russian: A comparative study. In *Proceedings of the International Conference Dialogue*.
- Forcada, M. L., Ginestí-Rosell, M., Nordfalk, J., O'Regan, J., Ortiz-Rojas, S., Pérez-Ortiz, J. A., Sánchez-Martínez, F., Ramírez-Sánchez, G., and Tyers, F. M. (2011). AperiTium: a free/open-source platform for rule-based machine translation. *Machine Translation*, 25(2):127–144.
- Gers, F. A., Schmidhuber, J. A., and Cummins, F. A. (2000). Learning to forget: Continual prediction with LSTM. *Neural Computation*, 12(10):2451–2471, October.
- Inoue, G., Shindo, H., and Matsumoto, Y. (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, Vancouver, Canada, August. Association for Computational Linguistics.
- Kanerva, J., Ginter, F., Miekka, N., Leino, A., and Salakoski, T. (2018). Turku neural parser pipeline: An end-to-end system for the CoNLL 2018 shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics.
- Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.
- Klyshinsky, E., Kochetkova, N., Litvinov, M., and Maximov, V. (2011). Method of POS-disambiguation using information about words co-occurrence (for Russian). *Proceedings of GSCL*, pages 191–195.
- Korobov, M. (2015). Morphological analyzer and generator for Russian and Ukrainian languages. In Mikhail Yu. Khachay, et al., editors, *Analysis of Images, Social Networks and Texts*, pages 320–332, Cham. Springer International Publishing.
- Koskenniemi, K. (1983). *Two-level morphology: A general computational model for word-form recognition and production*. Ph.D. thesis, Helsinki, Finland.
- Melamud, O., Goldberger, J., and Dagan, I. (2016). context2vec: Learning generic context embedding with bidirectional LSTM. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 51–61.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2, NIPS'13*, pages 3111–3119, USA. Curran Associates Inc.
- Moshagen, S. N., Pirinen, T., and Trosterud, T. (2013). Building an open-source development infrastructure for language technology projects. In *Proceedings of NODALIDA 2013: the 19th Nordic Conference of Computational Linguistics*, pages 343–352, Oslo, Norway.
- Nivre, J., Hall, J., Kübler, S., McDonald, R., Nilsson, J., Riedel, S., and Yuret, D. (2007). The CoNLL 2007 shared task on dependency parsing. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*.
- Nivre, J., Abrams, M., and al. (2018). Universal dependencies 2.3. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.
- Parra Escartín, C. and Martínez Alonso, H. (2015). Choosing a spanish part-of-speech tagger for a lexically sensitive task.
- Schmid, H. (1994). Probabilistic part-of-speech tagging using decision trees.
- Tkachenko, A. and Sirts, K. (2018). Neural morphological tagging for Estonian. *CoRR*, abs/1810.06908.
- Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Łukasz Kaiser, Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M., and Dean, J. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.
- Yatbaz, M. A. and Yuret, D. (2009). Unsupervised

morphological disambiguation using statistical language models. In *Pro. of the NIPS 2009 Workshop on Grammar Induction, Representation of Language and Language Learning*, Whistler, Canada, pages 321–324.

Zalmout, N. and Habash, N. (2017). Don't throw those morphological analyzers away just yet: Neural morphological disambiguation for Arabic. pages 704–713, 01.

A Examples of ambiguities

Additional examples of the kinds of ambiguities that our method handles (and does not handle):

We divide surface form ambiguities into three categories in the following subsections: two (or more) declinable lemmas, one declinable and one indeclinable lemma, or two indeclinable lemmas.

We classify lemmas into two types—depending on whether they accept inflectional morphemes: *declinable* lemmas accept them, and *indeclinable* lemmas do not. Thus, an indeclinable lemma has only one surface form. Declinable lemmas can have many surface forms.

In the examples, we use the following annotation convention:

- “the surface form”
- lemma
- (translation)
- POS and morphological tags

A1. Surface forms with two declinable lemmas

Different lemma, different POS:

Finnish surface form “tuli” has two readings:

FI “tuli”: (fire || *s/he came*)

tuli (fire) Noun, nominative, sing. ||

tulla (come) Verb, indicative, active, past tense, 3rd person, sing.

Russian surface form “стали” has two readings:

RU “стали”: (steel || *they became*)

сталь (steel) Noun, genitive, sing. ||

стать (become) Verb, indicative, active, past, 3rd person, plur.

Spanish surface form “vino” has two readings:

ES “vino”: (wine || *s/he came*)

vino (wine) Noun, sing. ||

venir (come) Verb, indicative, active, past, 3rd person, sing.

Same lemma, different POS:

Each of the following words (surface forms) has two readings, where the lemmas are the same, but the POS are different:

ES “parecer”:

parecer (seem), Verb, infinitive ||

parecer (opinion), Noun, sing.

RU “знать”:

знать (know) Verb, infinitive ||

знать (nobility) Noun, nominative, sing.

RU “стать”:

стать (become) Verb, infinitive ||

стать (posture) Noun, nominative, sing.

RU “печь”:

печь (bake) Verb, infinitive ||

печь (hearth) Noun, nominative, sing.

Different lemma, same POS:

This type of ambiguity is present in all languages. The following surface forms have two (or more) readings:

FI “pala”:

pala (returns) Verb, present, 3rd, sing. ||

palata (burns) Verb, present, 3rd, sing.

FI “alusta”:

alusta (pad, base) Noun, nominative, sing. ||

alus (ship) Noun, partitive, sing. ||

alunen (underlay) Noun, partitive, sing.

RU “черта”: (mark || *of the devil*)

черта (mark) Noun, nominative, sing. ||

черт (devil) Noun, genitive, sing.

RU “белка”: (squirrel (acc.) || *to the protein*)

белка (squirrel) Noun, accusative, sing. ||

белок (protein) Noun, dative, sing.

ES “fui”: (*I was* || *I went*)

ser (be) Verb, past perf., 1st, sing. ||

ir (go) Verb, past perf., 1st, sing.

Same lemma, same POS:

These are the kinds of ambiguities that our methods do not address, since both the lemma and POS are identical for the different analyses:

FI “nostaa”:

nostaa (raise), Verb, infinitive ||

nostaa (raise), Verb, present, 3rd, sing.

RU “кота”:

кот (cat) Noun, genitive, sing. ||

кот (cat) Noun, accusative, sing.

A2. Surface forms with one (or more) declinable and one indeclinable lemma

Different lemma, different POS:

ES “sobre”:

sobre (above) Preposition ||

sobre (envelope) Noun, sing. ||

sobrar (remain) Verb, present subjunctive, 1st/3rd, sing.

RU “уже”:

уже (already) Adverb ||

уж (grass snake) Noun, locative, sing. ||

узкий (narrow) Adj, comparative