Automatic Predicate Sense Disambiguation Using Syntactic and Semantic Features

Branko Žitko,[*] Lucija Bročić,[*] Angelina Gašpar,[†] Ani Grubišić,[*] Daniel Vasić,[‡] Ines Šarić-Grgić[*]

[*]Faculty of Science
University of Split
Ruđera Boškovića 33, 21000 Split, Croatia
branko.zitko@pmfst.hr, lucija.brocic@pmfst.hr, ani.grubisic@pmfst.hr, ines.saric@pmfst.hr

[†]Catholic Faculty of Theology
University of Split
Ulica Zrinsko Frankopanska 19, 21000 Split, Croatia
angelina.gaspar@kbf-st.hr

[‡]Faculty of Science and Education
University of Mostar
Poljička cesta 35, Mostar, Bosnia and Herzegovina
daniel.vasic@fpmoz.sum.ba

Abstract

This paper focuses on Predicate Sense Disambiguation (PSD) based on PropBank guidelines. Different approaches to this task have been proposed, from purely supervised or knowledge-based, to recently hybrid approaches that have shown promising results. We introduce one of the hybrid approaches - a PSD pipeline based on both supervised models and handcrafted rules. To train three supervised POS, DEP and POS DEP models we used syntactic features (lemma, part-of-speech tag, dependency parse) and semantic features (semantic role labels). These features enable per-token classification, which to be applied to unseen words, requires handcrafted rules to make predictions specifically for nouns in light verb constructions, unseen verbs and unseen phrasal verbs. Experiments were done on newly-developed dataset and the results show a token-level accuracy of 96% for the proposed PSD pipeline.

1. Introduction

One of the main tasks of Natural Language Processing (NLP) is precisely understanding the meaning of the word and its specific usage in a sentence, task known as Word Sense Disambiguation (WSD). In this paper, we focus on predicate sense disambiguation, i.e. the correct meaning of a predicate in a given sentence. A predicate combines with a subject to form a sentence, expressing some situation, event or state. Predicates are often single or compound verbs, consisting of various part-of-speech (prepositions, adverbs, nouns, auxiliaries, etc.). Hence, the precise understanding of the meaning of a sentence lies in the correct disambiguation of different types of words, not just verbs. For example, the term light verb (LV) refers to a verb that gets its main semantic content from the noun that follows rather than the verb itself. Thus, the construction consisting of such a verb and noun is called Light Verb Construction (LVC). In the sentence “I take a walk in the park.”, ‘take a walk’ is the LVC in which the noun ’walk’ describes an action. It is non-compositional and its lexical-syntactic structure is not flexible. This example illustrates that word sense disambiguation can make Predicate Sense Disambiguation (PSD) more accurate, since splitting up the LVC and disambiguating the senses of its components individually neglects the semantic unity of the construction and fails to represent its single meaning. Namely, ’walk’ can have a meaning of moving forward, one foot in front of the other, but it can also be a term specific for baseball. Depending on the sense of a word ’walk’, the sense of the whole predicate changes.

Another important role of PSD is the one it plays in Semantic Role Labelling (SRL). The process of semantic role labelling typically consists of predicate identification and its sense disambiguation, followed by identification of semantic roles and finally their labelling. The state-of-the-art BERT models like AllenNLP’s models (Gardner et al., 2018) or InVeRo (Conia et al., 2020) perform all mentioned subtasks except for predicate sense disambiguation which is missing. Ideally, the tool would use predicate senses to label semantic roles. However, we lack the tool for PSD, so we use the opposite technique – attempting to predict roleset IDs from already annotated semantic role labels. Another shortcoming of mentioned state-of-the-art models is that they only label verbs as predicates, and as we have seen, it is necessary to label words of different part-of-speech in addition to verbs. Regarding the sentence "I take a walk in the park.", state-of-the-art models identify word ’take’ as a predicate, whereas they ignore the word ’walk’. The need for such a PSD tool arises during the question generation task in intelligent tutoring system (Grubišić et al., 2020) our research team is working on.

In this work, we describe our PSD pipeline, depicted in Figure 1, as well as the process it takes to create it. The approach we take is the combination of the supervised PSD trained with the Stochastic Gradient Descent method (Kiefer and Wolfowitz, 1952) and the knowledge

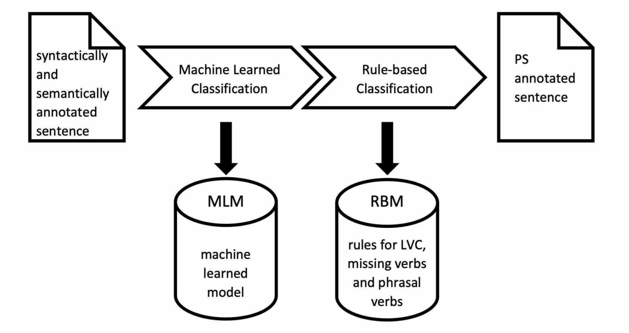
syntactically and semantically annotated sentence
Machine Learned Classification
Rule-based Classification
PS annotated sentence
MLM
machine learned model
RBM
rules for LVC, missing verbs and phrasal verbs

Figure 1: Our PSD Pipeline.

used to handcraft rules to compensate for the shortcomings of the data. We train supervised classifiers for each word to disambiguate senses based on extracted syntactic and semantic features, which play a significant role in many NLP tasks (e.g. text summarization, question generation, etc.). As for the syntactic features we use spaCy (Honnibal et al., 2020) annotated fine-grained POS (part-of-speech) tags and dependency tags. We employ the AllenNLP's BERT-based model (Gardner et al., 2018) to retrieve shallow semantics, represented by SRL labels. Thus, the proposed PSD pipeline consists of Machine Learned Classification (MLC) pipeline, based on Machine Learned Model (MLM), and Rule-Based Classification (RBC) pipeline, based on Rule-Based Model (RBM) including handcrafted rules for LVC, unseen verbs (verbs that don't occur in the OntoNotes dataset used for training the MLMs) and unseen phrasal verbs (phrasal verbs that don't occur in the OntoNotes dataset used for training the MLMs). We provide source code[1] with the spaCy integration of the proposed PSD pipeline.

Section 2 provides related works, which suggest that the WSD, which entails PSD, is a current problem encountered in various popular NLP tasks. Section 3 describes the dataset used for training PSD models and the modifications done to it. Section 4 describes the proposed PSD pipeline, providing detailed information on the training and evaluation of the models. Section 5 provides the conclusion of this paper and discussion about the given work.

2. Related Work

Word Sense Disambiguation and Predicate Sense Disambiguation are appealing NLP tasks for researchers in the field. Thus, they are the subject of many research activities, summarized in the up-to-date survey of recent trends in WSD (Bevilacqua et al., 2021). Among the various approaches to WSD, most popular are knowledge-based approaches, which often implement graph algorithms, and supervised approaches, which lately utilize neural networks.

Supervised WSD formulates the given task as classification task. Hence, it requires precisely labelled training data to learn the relationship between word annotations and senses. In contrast to a single classifier approach (Kawahara and Palmer, 2014), where one classifier is trained to make predictions for every word sense, there is also a per-verb approach (Chen and Eugenio, 2010). We implement the latter technique, where we train each classifier to disambiguate senses of only one word. Purely data-driven WSD is a straightforward approach when dealing with the comprehensive data. However, we find Supervised WSD approach that exploits relations between tokens more appealing. In that approach, some examples of improving the sense prediction might be by using contextual embeddings learned from Neural Language Model (Loureiro and Jorge, 2019), or by utilizing WordNet relations to create try-again mechanism to predict sense for ambiguous words (Wang and Wang, 2020).

On the other hand, knowledge-based WSD often implements various graph algorithms to extract from tokens and sentences their syntactic, semantic or any other features. These features are essential for modelling the Lexical Knowledge Base that algorithms use to predict senses. Although there are some high-scoring methods (Wang and Wang, 2020; Scozzafava et al., 2020) based on this approach, knowledge-based WSD systems still perform worse than supervised ones. However, lately there have been a few promising hybrid approaches that combine supervised and knowledge-based ones, as mentioned in the survey (Bevilacqua et al., 2021). Moreover, their high scores indicate that the hybrid approaches are currently the best solution to WSD (Barba et al., 2021). Besides the research done on WSD, there has also been some work concentrated specifically on Verb Sense Disambiguation (VSD). As verbal multiword expressions are semantically complex lexical items, there have been experiments to inspect the effect of the selection of semantic features in VSD. Research works like ours (Dang and Palmer, 2005; Dligach and Palmer, 2008; Ye and Baldwin, 2006) used SRL annotation, which is a distinctive characteristic of a predicate, to get better sense prediction.

3. Data Manipulation and Analysis

To build a good PSD model combining a supervised PSD approach and handcrafted rules, we need good data for the former and clear guidelines for sense disambiguation for the latter.

3.1. OntoNotes Data

We use an English corpus from the OntoNotes project as the train and test data for the supervised component of the model. The English dataset of the OntoNotes Release 5.0 (Weischedel et al., 2013) consists of 13109 annotated documents organized as .onf files, arranged into seven directories that correspond to files' sources. It is important to train the model on the content of assorted genres and types, therefore, OntoNotes was picked as it has the following seven categories: Broadcast Conversation (transcripts of talk shows from channels such as BBC, CNN and MSBNC), Broadcast News (news data collected from various news sources, such as ABC, NBC, CNN and Voice of America), Magazine (Sinorama Magazine), Newswire (data from sources such as Wall Street Journal newswire), Pivotal Corpus (biblical texts from the Old Testament and the New Testament), Telephone Conversation (conversational speech texts) and Web data (English web texts and

[1] https://github.com/lucijabrocic/PSD-pipeline

web text translated from Arabic and Chinese to English). The syntactic annotation of the sentences in the corpus followed the Penn TreeBank scheme and the predicate-argument structure followed the Proposition Bank (PropBank) annotation (Palmer et al., 2005). The OntoNotes English corpus consists of 143709 annotated sentences, most of which but not all have comprehensive annotation. Namely, some web texts selected to improve sense coverage were just tokenized and not even treebanked. Therefore, the corpus needed some refinement before further usage. The scripts (Bonial et al., 2014) provided by the Proposition Bank project enabled the conversion of original PropBank annotations (found in the OntoNotes project) to the new unified PropBank annotations. The files thus obtained were further modified by custom user-defined methods written for this work. Those methods mostly changed the aesthetics of the files, such as converting SRL annotation to utilize BIO notation and converting tree parses into dependency parse annotation. Finally, after the refinement and modifications, our corpus contains 7212 text files (137811 sentences), which follow the original OntoNotes directory structure based on files' sources.

3.2. The English PropBank

As already mentioned, the used data follows the English PropBank (Palmer et al., 2005) sense disambiguation guidelines. This research aims to predict the sense ID, also known as a frameset or roleset ID, for each word of any complex predicate structure in a sentence.

The English PropBank consists of 7311 .xml files called frame files, specifying the predicate-argument structure. One frame file, or frameset, consists of one predicate lemma or multiple different ones, and contains the information about roleset IDs that disambiguate various meanings of a predicate. Since diverse forms of a predicate can be under the same roleset ID, PropBank aliases can help to distinguish the correct sense from the wrong one. As our work required the English PropBank annotation information, we organized all the information for 10687 rolesets (and 7311 framesets) into easily loadable .json file.

No matter how large, representative, and carefully designed, no corpus can exhibit the same characteristics as a natural language. Having this in mind, we check the coverage of rolesets and framesets in the OntoNotes corpus. The analysis shows that the modified files miss 4922 rolesets and 3104 framesets, i.e. they cover 53.94% of rolesets and 57.54% of framesets that occur in the English PropBank. Even though the frequency of using missing framesets and rolesets might be low, the objective is to include as many framesets and rolesets as possible to increase the overall coverage. To achieve this objective, we add the handcrafted rules, explained more thoroughly in subsection 4.3.

4. The Proposed PSD Pipeline

This section describes the training process of three PSD models (POS, DEP and POS DEP) and their evaluation. We train each model by employing two approaches. In the first approach, we split the dataset into train and test sets, while in the second one, we use entire dataset for training.

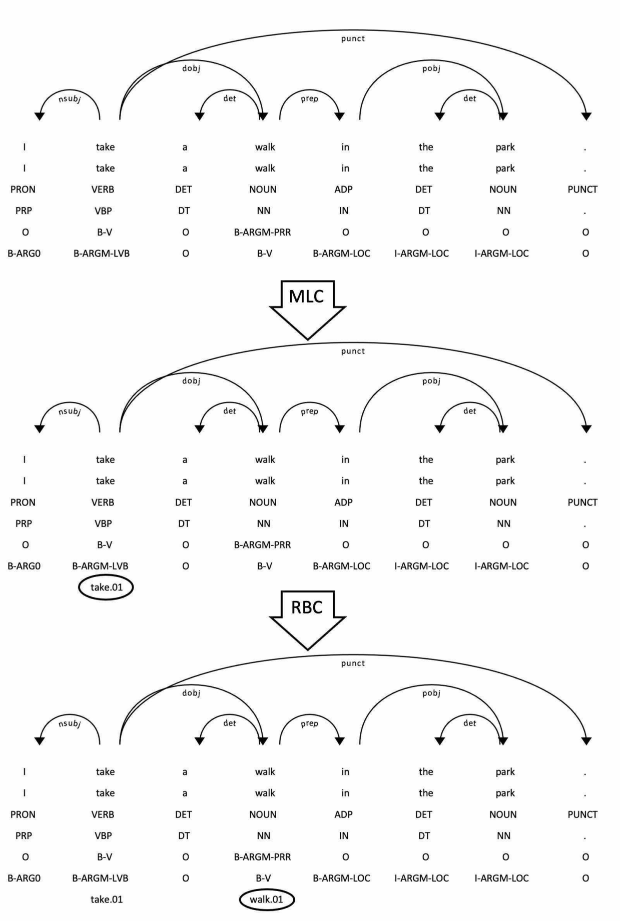
punct
dobj
pobj
nsubj
det
prep
det
I take a walk in the park .
I take a walk in the park .
PRON VERB DET NOUN ADP DET NOUN PUNCT
PRP VBP DT NN IN DT NN .
O B-V O B-ARGM-PRR O O O O
B-ARG0 B-ARGM-LVB O B-V B-ARGM-LOC I-ARGM-LOC I-ARGM-LOC O
MLC
punct
dobj
pobj
nsubj
det
prep
det
I take a walk in the park .
I take a walk in the park .
PRON VERB DET NOUN ADP DET NOUN PUNCT
PRP VBP DT NN IN DT NN .
O B-V O B-ARGM-PRR O O O O
B-ARG0 B-ARGM-LVB O B-V B-ARGM-LOC I-ARGM-LOC I-ARGM-LOC O
take.01
RBC
punct
dobj
pobj
nsubj
det
prep
det
I take a walk in the park .
I take a walk in the park .
PRON VERB DET NOUN ADP DET NOUN PUNCT
PRP VBP DT NN IN DT NN .
O B-V O B-ARGM-PRR O O O O
B-ARG0 B-ARGM-LVB O B-V B-ARGM-LOC I-ARGM-LOC I-ARGM-LOC O
take.01 walk.01

Figure 2: The syntactically and semantically annotated sentence "I take a walk in the park." enters MLC pipeline, which annotates the predicate sense for verb "take" as take.01. The annotated sentence then proceeds to the RBC pipeline, which annotates the predicate sense for noun "walk" as walk.01.

Figure 2 illustrates the PSD pipeline with an example input sentence, annotated with syntactic and semantic features. First the MLC pipeline extracts these features from the sentence and feeds them to the trained classifiers used to obtain predicate senses. Then, RBC pipeline takes the syntactically and semantically annotated sentence with predicted predicate senses. RBC pipeline applies handcrafted rules to the sentence to improve the prediction of predicates in light verb constructions, unseen verbs and unseen phrasal verbs. As a result of the proposed pipeline processing, each token in the sentence has a roleset attribute that stores the result.

4.1. Training the Models

We have 7212 OntoNotes files available to make the best use of while training our models. We first apply a typical supervised learning approach - splitting the dataset into the train and test sets and then performing the training and evaluation. The train-test split given in the PropBank (Bonial et al., 2014) resulted in 80% of the files (and sentences) in train set and 20% in the test set.

Table 1 shows that many framesets and roleset IDs oc-

	No. of files	No. of sentences	No. of framesets	No. of roleset IDs
Train set	5832	111104	3996	5455
Test set	1380	26707	2692	3609
Corpus	7212	137811	4208	5766

Table 1: Corpus composition.

cured in both train and test set. Out of 2692 framesets identified in the test set, 212 framesets did not appear in the train set. Likewise, out of 3609 roleset IDs detected in the test set, 311 of them failed to appear in the train set.

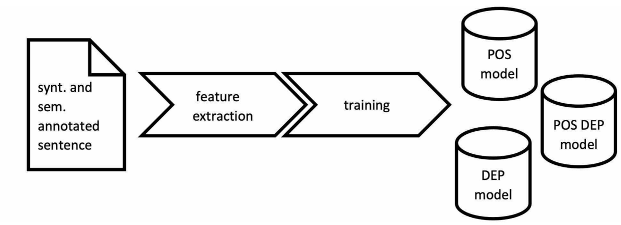

Figure 3: The models' training pipeline.

Figure 3 illustrates the training process. First, the syntactically and semantically annotated sentence is loaded and forwarded to feature extraction.

During the feature engineering and extraction phase, the most relevant token-level annotations for developing the models are selected. Those annotations are token text, its modified lemma that matched the English PropBank frameset, part-of speech (POS) tag, dependency parse and semantic role labels (SRL). The research (Dang and Palmer, 2005; Dligach and Palmer, 2008) shows that the predicate sense disambiguation could improve semantic role labelling. Ideally, word sense disambiguation would solve the problem of identifying the correct sense of a polysemic word based on context. However, the lack of comprehensive repository of senses and a tool for PSD prompted us to use the opposite technique - attempting to predict roleset IDs from already annotated semantic role labels. As for the POS and dependency annotation, previous studies show the performance of the SRL task heavily depends on the performance of dependency parsing (Mohammadshahi and Henderson, 2021) and POS tagging (Wilks and Stevenson, 1997) subtasks. We train three models and name them according to the features they used - POS, DEP and POS DEP. All three models utilize token text and lemma, but differ in the other used annotation(s): (i) the POS model utilizes the relation between SRL and fine-grained POS tag, (ii) the DEP model utilizes the relation between SRL and dependency tag, (iii) the POS DEP model utilizes the relation between SRL, fine-grained POS tag and dependency tag. In this research, we train and evaluate the three models in parallel.

To be more specific, we present featuresets of tokens "take" and "walk" in the Figure 2 used when employing the POS DEP model. Token "take" has only one SRL argument - token "walk" which is ARGM-PRR. On the other hand, token "walk" has three SRL arguments - token "I" that is ARG0, token "take" that is ARGM-LVB, and finally tokens "in", "the" and "park" that are ARGM-LOC. Therefore, the featureset for token "take" is $\big[\big(\text{text, take}\big), \big(\text{lemma, take}\big), \big(\text{ARGM-PRR}, \big[\langle\text{NN, dobj}\rangle\big]\big)\big]$, and for token "walk" $\big[\big(\text{text, walk}\big), \big(\text{lemma, walk}\big), \big(\text{ARG0}, \big[\langle\text{PRP, nsubj}\rangle\big]\big), \big(\text{ARGM-LVB}, \big[\langle\text{VBP, ROOT}\rangle\big]\big), \big(\text{ARGM-LOC}, \big[\langle\text{IN, prep}\rangle, \langle\text{DT, det}\rangle, \langle\text{NN, pobj}\rangle\big]\big)\big]$.

Then we vectorize extracted features and feed them into the classifiers. Dealing with PSD, we face a multiclass classification problem with more than 10000 classes. Instead of a single classifier, a common solution to a problem like this is training multiple binary classifiers, one for each class of the original problem. In the NLP-like domains, however, it is more suitable to use multiple classifiers which predict a constricted number of classes (Even-Zohar and Roth, 2001). Therefore, in this research, multiple multiclass classifiers perform the classification task, with one classifier for each frame file. Hence, the number of classifiers auguments to 7311, and each has to learn the nuances between roleset IDs within the same frame file. The model itself is essentially a collection of such classifiers.

Regarding the choice of classifier, we want to build a simple and fast model for this PSD task. Since the context we need is already assigned to a token through context-aware models (spaCy, AllenNLP), with some feature engineering we can utilize generated annotations (lemma, POS, dependency, SRL) as features for our model. Hence, we did not take a neural approach, but we decided on a linear classifier where learning is based on multinominal logistic regression with SGD optimization.

4.2. The evaluation of models' accuracy and performance

We evaluate our models on the OntoNotes test set containing 26707 sentences. Those sentences contain in total 504891 tokens, of which 75621 (or 14.98%) are predicate tokens, and 429270 (or 85.02%) are non-predicate tokens. When looking at the average sentence, it contains 18.90 tokens, of which 2.83 are predicate tokens and 16.07 are non-predicate tokens. We measure the accuracy of the three PSD models on this OntoNotes test set with three different metrics:

- the token-level accuracy (TLA) metric measures the number of (predicate and non-predicate) tokens the model predicted correctly (correct roleset ID or no prediction, depending on whether the token is a part of a predicate or not)
- the sentence-level accuracy (SLA) metric measures the number of sentences the model predicted completely correctly (all the tokens)
- the predicate-level accuracy (PLA) metric measures the number of predicate tokens the model predicted correctly

Besides accuracy, we also use predicate prediction coverage (PPC) metric, which represents the ratio of predicted predicate tokens and total predicate tokens (whether they are correctly predicted or not). When evaluating AllenNLP's BERT model on OntoNotes test set, we can obtain a measure similar to PPC. Looking at the ratio between predicate tokens in OntoNotes test set for which AllenNLP annotates the SRL arguments and all predicate tokens in OntoNotes test set, we get a result of 88.02%. It is important to note that the remaining 11.98% are nouns for which AllenNLP's BERT model cannot annotate SRL labels. This coverage metric for AllenNLP puts into perspective the PPC measure of our models, given in Table 2.

	TLA (%)	SLA (%)	PLA (%)	PPC (%)
POS	98.50	76.91	90.01	97.49
DEP	98.71	79.74	91.37	97.82
POS DEP	98.73	80.04	91.54	97.97

Table 2: Evaluation of the Models.

Table 2 shows that the results of evaluation metrics on accuracy are similar for the three models, even though POS DEP model is the most accurate and obtained the highest PPC score. As explained in subsection 4.1., models encounter some framesets and roleset IDs in the test set alone. After the initial training and evaluation phase, we further train models on all 7212 modified OntoNotes files, assuming their performance would improve. To distinguish which results correspond to which model, we will use two terms: OntoNotes-split model and OntoNotes-whole model. The term OntoNotes-split model will denote model that is trained on OntoNotes train set and evaluated on OntoNotes test set, while OntoNotes-whole model will denote model that is trained on all of the 7212 OntoNotes files. The results given so far are for OntoNotes-split models.

4.3. PSD Pipeline

Even when trained on all available data, our PSD models cover only 53.94% of rolesets and 57.54% of framesets in the the English PropBank. Therefore, we handcraft rules to improve the predictive abilities of models.

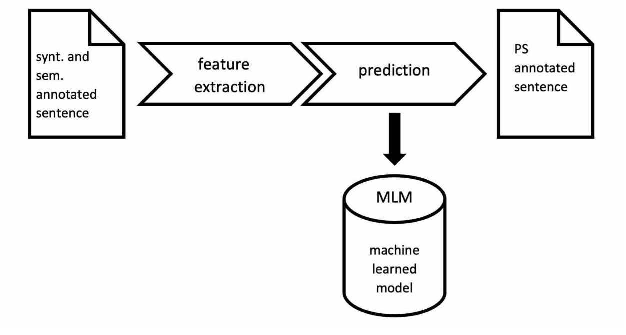

Figure 4: The MLC component of the PSD Pipeline.

Figure 4 presents the Machine Learned Classification (MLC) component of the PSD pipeline, which uses the ML model to make a predicate sense prediction. In model training phase, we use the OntoNotes annotation of sentences for feature extraction. However, when using the PSD pipeline "in the wild" on arbitrary sentences, spaCy's English RoBERTa-based transformer processing pipeline uses the raw input to retrieve syntactic features. The AllenNLP's BERT model is used to obtain semantic features, added to spaCy objects (Token, Span, Doc) via the custom SRL pipe. One thing to note is that we slightly modify both the spaCy pipeline and AllennNLP's BERT model. We improve spaCy's lemmatizer to better lemmatization of gerunds and contracted verbs. The modifications made to the AllenNLP's BERT model allow the presence of nouns in a predicate and adjustment of SRL labels for LVCs to the English PropBank guidelines.

Next, syntactic and semantic features are extracted in the same way as it has been described in the training phase (Subsection 4.1.). The prediction can be done using one of the three previously mentioned OntoNotes-whole models (POS, DEP, POS DEP), and each model is essentially a collection of classifiers that each corresponds to a Penn PropBank frameset. The output of MLC component is a sentence where predicate tokens are annotated with sense predicted via classifiers.

Figure 5 illustrates further processing of annotated sentences in the Rule-Based Classification (RBC) component based on the Rule-Based Model, including handcrafted rules for LVC, unseen verbs and unseen phrasal verbs to improve prediction.

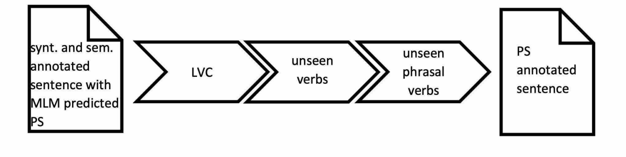

Figure 5: The RBC component of the PSD Pipeline.

Essentially, a sentence with classifier-predicted PSD annotation is forwarded to the RBC pipeline component to first handle the sense disambiguation of nouns in LVCs. Then the RBC component uses modified SRL labels to find both parts of an LVC and search for PropBank aliases to find the corresponding one. The pipeline component explores aliases labeled as nouns only if there are no aliases tagged as the light verbs. This way PropBank aliases help in finding the correct sense IDs.

The next step includes the sense disambiguation of unseen verbs. The RBC component searches for PropBank aliases tagged as verbs, attempting to find the potential sense (roleset ID) of verbs that do not occur in the training set.

In the last step, the pipeline component performs the sense disambiguation of two-word phrasal verbs. Phrasal verbs are easy to predict correctly using the rules. The RBC pipeline first checks if a verb has a dependant particle (eg. a preposition or an adverb) and searches the PropBank aliases tagged as verbs, to find a corresponding sense (roleset ID).

The RBC pipeline makes prediction in each step only if the observed token (i) has SRL labels (AllenNLP model identified the token as predicate), (ii) is not a modal verb (no sense disambiguation of modals) and (iii) has no prediction (the goal is to supplement the classifiers, not to overwrite their predictions).

Moreover, we introduce new annotations in the three steps of the RBC pipeline. For a better understanding, examples in Table 5 illustrate predictions of the PSD pipeline components using the POS DEP model and their possible outcomes. The search for PropBank aliases can result in a lack of roleset ID matches, only one roleset ID match, or multiple roleset ID matches. Table 5 shows how each pipeline component resolves the roleset ID issue depending on the number of found rolesets matches.

When there is no corresponding roleset ID for the token, the actions of the RBC pipeline differ based on the predicate construction. If the token is a part of an LVC (e.g. picnic - None), the RBC pipeline predicts the sense disambiguation as the lemma of the token followed by ".00" (picnic - picnic.00). If the token is an unseen verb (e.g. overwrite - None) or a part of an unseen phrasal verb (e.g. clue - None), however, the sense remains unchanged (None).

If there is only one roleset ID match, components of the RBC pipeline choose that roleset ID.

If there are multiple roleset ID matches, components of the RBC pipeline choose the roleset ID with the lowest number, followed by the flag "X". However, this annotation indicates that the unique prediction is still not achievable.

Finally, our PSD pipeline incorporates final sense prediction into the spaCy's processing pipeline, into custom roleset attribute.

5. Experimental Results and Discussion

This section provides the results obtained on the gold standard dataset and discussion and suggestions for further work.

5.1. The Evaluation of the Model Performance on the Gold Standard Dataset

As all three OntonoNotes-split models perform similarily well, we further assess the accuracy of the OntoNotes-whole POS DEP model on a fresh set of sentences that represent our gold standard. The new dataset consists of manually annotated 664 sentences with syntactic (lemmatization, part-of-speech and dependency tags) and semantic (SRL) labels, and the predicate sense IDs which our model predicts. In Table 3 are given statistics for the dataset considering tokens, words and predicates. Tokens include both words and non-word parts of a sentence, e.g. punctuation. When expressed as a percentage, 18.46% tokens in the gold dataset are predicates.

	total	per sentence			
		mean	std	min	max
token	6853	10.320	4.770	2	65
word	5971	8.992	3.430	1	48
predicate	1265	1.905	0.890	0	12

Table 3: Gold dataset statistics.

The first step of evaluation process includes the predicate sense prediction using input sentence and the needed annotations obtained through system (spaCy transformer model and AllenNLP's BERT model) pipeline. In the second step, as some system annotations are erroneous, namely, wrong lemmatization and SRL labels, we use gold standard annotations to check if there is any difference in prediction.

	TLA (%)		**SLA (%)**		**PLA (%)**		**PPC (%)**
	X	no X	X	no X	X	no X	X & no X
pipeline	96.19	92.20	69.28	69.43	87.11	87.17	98.05
gold standard	97.63	97.67	78.01	78.46	89.75	89.94	100.00

Table 4: Evaluation of the POS DEP model on the gold standard dataset.

The evaluation results in Table 4 show that the OntoNotes-whole POS DEP model predicts better if fed with human-made annotations rather than with system-generated annotations. The most significant difference is in sentence-level accuracy, resulting from higher token-level and predicate-level accuracies.

To put the PPC measure given in Table 4 in perspective, we evaluate AllenNLP's BERT model on the gold standard dataset and obtain a measure similar to PPC. Looking at the ratio between predicate tokens in the dataset for which AllenNLP annotates the SRL arguments and all predicate tokens in the dataset, we get a result of 97.61%. When using system-generated annotations, our OntoNotes-whole POS DEP model relies on AllenNLP for discovering the predicates it needs to predict senses for. By deeper analysis, it is visible that there are certain errors in spaCy's system-generated annotations (namely lemma) that lower the original AllennNLP coverage of 97.61%. However, the modifications made to the AllenNLP's BERT model that allow presence of nouns in a predicate have increased our predicate coverage of 98.05%, and in the end improved the original AllenNLP's coverage of 97.61%.

The POS DEP model returns rolesets with “X” flag when it cannot decide between multiple different senses. To fully evaluate the model's performance, we calculated the four metrics on the predictions with removed "X" flag (no X). The slight increase in scores indicates that the roleset with the lowest ID number was often the right one.

5.2. Discussion and Further Work

We have shown our approach to predicate sense disambiguation utilizing POS, dependency and SRL annotations, and on the way presented the analysis of the coverage of the predicate senses in the OntoNotes corpus and the English PropBank contrastively. The integration of PSD pipeline into spaCy makes its usage straightforward - by adding a custom SRL and roleset components to the spaCy processing pipeline.

Another feature of the proposed PSD pipeline is its Machine Learned Models (MLMs). Each model consists of per-token classifiers, which implies some effort required to combine their outputs. However, the predicate sense prediction is fast since the pipeline only employs the classifiers corresponding to framesets found in the sentence. Moreover, changing the single classifier is simplified – if there is a change in annotation guidelines within one frame file, only one smaller classifier requires retraining. We have also presented different accuracy and prediction metrics used in evaluation of models' performance.

The scores in Table 4 suggest our PSD pipeline ob-

		LVC	**Unseen verbs**	**Unseen phrasal verbs**
Roleset ID doesn't exist	**Sentence**	Let's have a picnic in the park.	It will overwrite the files on your hard drive.	She'll clue you in on the latest news.
	MLC prediction	have – have.01 picnic - None	overwrite - None	clue - None
	MLC + RBC prediction	have – have.01 picnic - picnic.00	overwrite - None	clue - None
Unique roleset IDs exist	**Sentence**	He is having an affair.	Some people annotate as they read.	The cat scrunched up to sleep.
	MLC prediction	is – be.03 having - have.01 affair - None	annotate – None read – read.01	scrunched – None
	MLC + RBC prediction	is – be.03 having - have.01 affair – affair.01	annotate – annotate.01 read – read.01	scrunched – scrunch_up.01
Multiple roleset IDs exist	**Sentence**	We are making a plea to all companies.	John frowned when he heard the news.	They sluice the streets down every morning.
	MLC prediction	are – be.03 making – make.01 plea - None	frowned – None heard – hear.01	sluice - None
	MLC + RBC prediction	are – be.03 making – make.01 plea - plead.01X	frowned – frown.01X heard – hear.01	sluice – sluice_down.01X

Table 5: Examples for PSD pipeline.

tains satisfactory results, however, there is still room for improvement. More specifically, in our further work, we plan to enhance the Rule-Based Classification (RBC) component, particularly sense disambiguation of unseen words with multiple rolesets based on their part-of-speech tags. The PSD pipeline only chooses the roleset with the lowest roleset ID and adds the flag “X”. We assume we can achieve better results if we create a more complex rule, as the one that utilizes PropBank guidelines on roleset sense IDs and their corresponding arguments in predicate-argument structure. Since there is a large number of missing rolesets and framesets (46.06% and 42.46% respectively), that will be no easy task and more in-depth analysis is necessary to figure out what mistakes does the model make and how to fix them.

We build our Rule-Based Models (RBMs) on three categories of words – nouns in Light Verbs Construction (LVC), unseen verbs and unseen phrasal verbs. Perhaps categories could be further disambiguated and thus, enable a better RBM. Another change that might be beneficial for improving the results is a selection of more features during the feature extraction phase. For a certain predicate, we use only POS and dependency tags of its arguments, but the accuracy might improve if we consider the text of the argument token as well.

Finally, the downstream task this PSD pipeline is created for is the question generation task in our intelligent tutoring system. Disambiguating predicate senses and capturing information about its arguments and characteristics will be useful when deciding on appropriate wh-word in a question.

Acknowledgements

The presented results are the outcome of the research project “Enhancing Adaptive Courseware based on Natural Language Processing (AC&NL Tutor)” undertaken with the support of the United States Office of Naval Research Grant (N00014-20-1-2066).

6. References

Edoardo Barba, Tommaso Pasini, and Roberto Navigli. 2021. ESC: Redesigning WSD with extractive sense comprehension. In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4661–4672, Online. Association for Computational Linguistics.

Michele Bevilacqua, Tommaso Pasini, Alessandro Raganato, and Roberto Navigli. 2021. Recent trends in word sense disambiguation: A survey. In: Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 4330–4338. International Joint Conferences on Artificial Intelligence Organization.

Claire Bonial, Julia Bonn, Kathryn Conger, Jena D. Hwang, and Martha Palmer. 2014. PropBank: Semantics of new predicate types. In: *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3013–3019, Reykjavik, Iceland. European Language Resources Association (ELRA).

Lin Chen and Barbara Di Eugenio. 2010. A Maximum Entropy Approach To Disambiguating VerbNet Classes. In: *Proceedings of Verb 2010, 2nd Interdisciplinary Workshop on Verbs, The Identification and Representation of Verb Features*.

Simone Conia, Fabrizio Brignone, Davide Zanfardino, and Roberto Navigli. 2020. InVeRo: Making semantic role labeling accessible with intelligible verbs and roles. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 77–84, Online. Association for Computational Linguistics.

Hoa Trang Dang and Martha Palmer. 2005. The role of semantic roles in disambiguating verb senses. In: *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 42–49, Ann Arbor, Michigan. Association for Computational Linguistics.

Dmitriy Dligach and Martha Palmer. 2008. Novel semantic features for verb sense disambiguation. In: *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, HLT-Short '08, page 29–32, USA. Association for Computational Linguistics.

Yair Even-Zohar and Dan Roth. 2001. A sequential model for multi-class classification. In: *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A deep semantic natural language processing platform. In: *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6, Melbourne, Australia. Association for Computational Linguistics.

Ani Grubišić, Slavomir Stankov, Branko Žitko, Ines Šarić-Grgić, Angelina Gašpar, Suzana Tomaš, Emil Brajković, and Daniel Vasić. 2020. Declarative Knowledge Extraction in the AC&NL Tutor. In: Robert A. Sottilare and Jessica Schwarz, editors, *Adaptive Instructional Systems*, pages 293–310, Cham. Springer International Publishing.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Boyd Adriane. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Daisuke Kawahara and Martha Palmer. 2014. Single classifier approach for verb sense disambiguation based on generalized features. In: *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 4210–4213, Reykjavik, Iceland. European Language Resources Association (ELRA).

Jack Kiefer and Jacob Wolfowitz. 1952. Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics*, 23(3):462–466.

Daniel Loureiro and Alípio Jorge. 2019. Language modelling makes sense: Propagating representations through WordNet for full-coverage word sense disambiguation. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5682–5691, Florence, Italy. Association for Computational Linguistics.

Alireza Mohammadshahi and James Henderson. 2021. Syntax-aware graph-to-graph transformer for semantic role labelling.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

Federico Scozzafava, Marco Maru, Fabrizio Brignone, Giovanni Torrisi, and Roberto Navigli. 2020. Personalized PageRank with syntagmatic information for multilingual word sense disambiguation. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 37–46, Online. Association for Computational Linguistics.

Ming Wang and Yinglin Wang. 2020. A synset relation-enhanced framework with a try-again mechanism for word sense disambiguation. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6229–6240, Online. Association for Computational Linguistics.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, Mohammed El-Bachouti, Robert Belvin, and Ann Houston. 2013. OntoNotes Release 5.0.

Yorick Wilks and Mark Stevenson. 1997. The grammar of sense: Using part-of-speech tags as a first step in semantic disambiguation. *Natural Language Engineering*, 4.

Patrick Ye and Timothy Baldwin. 2006. Verb sense disambiguation using selectional preferences extracted with a state-of-the-art semantic role labeler. In: *Proceedings of the Australasian Language Technology Workshop 2006*, pages 139–148, Sydney, Australia.