Argument Identification and Classification in Semantic Role Labeling

Group 13

Selin Acikel	Murat Ertas	Tessel Haagen	Csenge Szabó
2637714	2775481	2826310	2803781

Github repository: https://github.com/asa-sel/Advanced-NLP-13.git

1 Introduction

The purpose of this report is to delve into the identification and classification of arguments and adjuncts in Semantic Role Labeling (SRL). This is vital for extracting meaning from text data as it helps determine the roles and properties of participants in a sentence, including the ”who,” ”what,” ”whom,” ”when,” ”where,” and ”how” of the actions being described.

Our focus will be on identifying and classifying arguments and adjuncts to predicates, assuming that the predicate information is already provided. This will include the classification of their semantic roles, such as agent, patient, or instrument. We are utilizing the Universal Proposition Bank (v1.0) dataset for training a Logistic Regression classifier.

The report will provide an overview of the task and the pipeline in Section 2, introduce related work in Section 3, describe the dataset and the preprocessing steps in Section 4, explain the conducted feature-based experiments in Section 5 and analyze the results in Section 6. Our goal is to give a comprehensive understanding of the SRL task's implications for advancing NLP research and applications.

2 Task Description

In their work, Carreras and Màrquez (2005) introduce Semantic Role Labeling (SRL) as the objective of the CoNLL-2005 shared task. Extracting semantic information from textual data is essential for several downstream Natural Language Processing (NLP) tasks, including question-answering, information extraction, and machine translation. The aim of SRL is to determine the properties and participants of an event by answering the questions ”who” did ”what” to ”whom”, ”when”, ”where”, and ”how”. SRL is a complex sequence labeling task composed of multiple sub-tasks:

- Predicate identification: determining which tokens introduce an event or proposition in a sentence,
- Predicate classification: disambiguating the role set of the predicates
- Argument identification: determining which tokens are arguments or adjuncts belonging to the same predicate within a sentence,
- Argument classification: classifying the semantic role of arguments (e.g., agent, patient, instrument) and adjuncts (e.g., locative, temporal, manner).

Our paper only focuses on the task of argument identification and classification, and presumes information about predicates provided in the dataset as gold. *Proposition* refers to the combination of the governing predicate and its group of arguments and adjuncts, thus one sentence may be composed of multiple propositions depending on the number of predicates (Gildea and Jurafsky, 2002). Predicates are not limited to verbs but may also include nouns and adjectives. The examples below illustrate the task of identifying and determining the role of each argument, separated by square brackets, that belong to the same predicate.

(1) [*ARG0* John] **opened** [*ARG1* the door] [*ARG2* with his foot].
(2) [*ARG1* The door] **opened**.

In this sentence the verb 'open.01' is the predicate, and the semantic roles of its arguments are agent (ARG0), the thing opening (ARG1), and the instrument (ARG2) in the first sentence. In the second sentence, the role of the first constituent is not agent-like but rather patient-like. This example of *causative-inchoative alternation* shows that syntactic structure by itself is insufficient for solving the complex task at hand.

It is worth noting the arrangement of the arguments in a given proposition. Arguments do not overlap and cannot be repeated for the same predicate. Moreover, arguments may stretch across multiple non-continuous phrases or may be referential, as in the sentence '[ARGM-LOC The bed] on [R-ARM-LOC which] [ARG0 I] **slept** broke.' (Punyakanok et al., 2008).

2.1 Pipeline Description

The sub-tasks argument identification and classification may be carried out in two steps with separate classifier heads or combined into a single step. Due to the limited time and resources available for our experiments, we decided to choose the second approach with the help of supervised machine learning, acknowledging that higher scores might be achieved with the two-step approach. Given a set of possible argument classes and extracted features, including the predicate itself, the model is expected to predict if the tokens in the input sentence belongs to one one of the argument classes or is not an argument. In the same step, the model is expected to choose the most likely semantic role for each argument candidate.

3 Related work

Studies on SRL have been implemented through various methodologies, from early rule-based models to feature-based statistical models, and recently to the latest neural network approaches. In this section we provide a brief overview of the most significant landmarks in the evolution of SRL research.

One of the prominent early works for SRL was conducted by Gildea and Jurafsky (2000), who utilized statistical methods alongside syntactic features to predict the predicate-argument structure of sentences. Their work highlighted the importance of syntactic constituency trees, proving them to be invaluable for predicting predicate-argument relationships.

Thanks to availability of large annotated resources such as PropBank (Palmer et al., 2005), many SRL experiments have relied heavily on lexical and syntactic features and achieved high performance (Roth and Lapata, 2016). As an example, Pradhan et al. (2005) used the Support Vector Machine algorithm with an extensive set of lexical-syntactic features in their research, making an elaborate use of constituency parsing. They achieved state-of-the-art results with their model for its time.

The shift towards dependency-based approaches marked a significant evolution in SRL. These methods, gaining popularity in the CoNLL 2008 and 2009 shared tasks (Surdeanu et al., 2008; Hajič et al., 2009), analyze the dependencies between words in a sentence to assign semantic roles. (Marcheggiani et al., 2017) Utilizing dependency features has been demonstrated to be useful in improving predictive capacities for various models with respect to capturing relations between predicates and arguments. This significant improvement indicates that dependency relations serve as highly informative syntactic features for identifying semantic relationships between words (Hacioglu, 2004; Shi et al., 2020; Zhang et al., 2018; Roth and Lapata, 2016).

Most of the previously mentioned works rely on extensive feature engineering making use of complex sets of lexical-syntactic features. In multilingual systems, where extensive feature engineering is not practical, neural models have gained traction as a viable alternative that eliminates the need for hand-crafted features for SRL (Gesmundo et al., 2009; Henderson et al., 2013). The neural models for SRL even outperformed their traditional machine learning counterparts on standard benchmarks for English. (Marcheggiani et al., 2017)

4 Dataset Description

For this project we are utilizing the Universal Proposition Banks version 1.0[1] for English language, which was created with the aim to study cross-lingual semantics. The dataset - composed of mainly news articles, emails and reviews - was annotated with predicate-argument relations, semantic role labels and provides syntactic dependency information about the sentence structure (Palmer et al., 2005). Table 1 displays an example of the CoNNL-U-Plus formatted data. Sentences are separated by newlines in the files, the sentence ID and the full sentence are encoded on top of each sentence block. Each token of the sentence is introduced on a separate line, and the columns are separated by tabs. The example sentence has two predicates: be.01 and see.01, where the numbers refer to the role set ID of the verb.

ID	Token	Lemma	POS-UNIV	POS	Morph.	Head	BasicDep	EnhancedDep	Misc.	Predicate	Labels P1	Labels P2
1	It	it	PRON	PRP	Case=(...)	4	expl	4:expl	_	_	_	_
2	was	be	AUX	VBD	Mood=(...)	4	cop	4:cop	_	be.01	V	_
3	very	very	ADV	RB	_	4	advmod	4:advmod	_	_	_	_
4	upsetting	upsetting	ADJ	JJ	Degree=Pos	0	root	0:root	_	_	ARG2	_
5	to	to	PART	TO	_	6	mark	6:mark	_	_	_	_
6	see	see	VERB	VB	VerbForm=(...)	4	csubj	4:csubj	_	see.01	ARG1	V
7	this	this	DET	DT	Number=(...)	8	det	8:det	_	_	_	_
8	kind	kind	NOUN	NN	Number=(...)	6	obj	6:obj	_	_	_	ARG1
9	of	of	ADP	IN	_	10	case	10:case	_	_	_	_
10	behavior	behavior	NOUN	NN	Number=(...)	8	nmod	8:nmod:of	_	_	_	_
11	especially	especially	ADV	RB	_	13	advmod	13:advmod	_	_	_	_
12	in	in	ADP	IN	_	13	case	13:case	_	_	_	_
13	front	front	NOUN	NN	Number=(...)	6	obl	6:obl:in	_	_	_	ARGM-LOC
14	of	of	ADP	IN	_	17	case	17:case	_	_	_	
15	my	my	PRON	PRP$	Number=(...)	17	nmod:poss	17:nmod:poss	_	_	_	_
16	four	four	NUM	CD	NumType=(...)	17	nummod	17:nummod	_	_	_	_
17	year	year	NOUN	NN	Number=(...)	13	nmod	13:nmod:of	_	_	_	_
18	old	old	ADJ	JJ	Degree=Pos	17	amod	17:amod	SpaceAfter(...)	_	_	_
19	.	.	PUNCT	.	_	4	punct	4:punct	_	_	_	_

Table 1: Example Sentence from the Training Set

There are 10 main columns in the files, which contain the following information: (1) token ID, (2) token, (3) lemma, (4) universal part-of-speech (POS) tag, (5) POS tag, (6) morphological features, (7) dependency head of the token, (8) basic dependency relation, (9) enhanced dependency relation, (10) miscellaneous information. If the sentence has predicates, column 11 contains the predicate-sense labels or "_" otherwise. Columns 12th to nth vary in size depending on the number of predicates in the sentence. If the sentence is without predicates, it contains only 11 columns. If the sentence has a single predicate, the argument structure of the predicate is provided in column 12. In case of two predicates per sentence, the arguments in column 12 correspond to the first predicate and arguments in column 13 correspond to the second predicate, and so on. In this dataset only the head words of the argument were labeled with argument roles.

Argument	Description
ARG0	agent
ARG1	patient
ARG2	instrument, benefactive, attribute
ARG3	starting point, benefactive, attribute
ARG3	ending point
ARGM	modifier

Table 2: List of arguments in PropBank adapted from (Bonial et al., 2012)

In the next step, we investigated the argument distribution within the dataset. There are more negative samples (non-argument tokens) than positive samples (argument tokens), which stems from the nature of SRL. There is also a noticeable imbalance in the labels' distributions, see Figure 2 in the Appendix. In the statistics continuous arguments ('C-') and referential arguments ('R-') were merged with regular arguments, but they are handled as separate classes during evaluation. The most frequent argument types are ARG0 which stands for agent-like and ARG1 for patient-like or theme-like arguments across all

[1] https://universalpropositions.github.io

sets. Certain argument categories, such as ARG5, ARGA and ARGM-REC are underrepresented in the training data, therefore, we expect that the model might struggle with learning to correctly predict these categories. Table 2 summarises how the numbered arguments correspond to various semantic roles in the PropBank annotation.

4.1 Preprocessing

Before extracting features for our model, we preprocessed the sentences in our data. Sentences that contained multiple predicates were duplicated depending on the number of predicates, and the corresponding arguments and adjuncts were stored separately. As a result, all sentences with a predicate are composed of 12 columns, thus the argument labels in the final column correspond with the given predicate. Sentences containing no predicates or having fewer than 11 columns in the dataset were removed during preprocessing, which accounts for 1990 sentences in the training, 467 sentences in the development and 540 sentences in the test set. The reason for this is that sentences of this kind contain no arguments either due to the lack of predicates.[2] Finally, we replaced the argument labels marking the predicates ('V' and 'C-V') with '_' in column 12, since the objective of this project is exclusive to the identification and classification arguments and adjuncts. Table 3 below contains statistical information about the distributions in the dataset before and after preprocessing. The dataset is split into training, development and test set.

	Training	**Dev.**	**Test**	**P. Training**	**P. Dev.**	**P. Test**
# Sentences	12.543	2002	2077	40.482	4977	4799
# Tokens	204.609	25.150	25.097	1.038.137	105.068	101.144
# Sentences without predicates	1990	467	540	0	0	0
# Sentences with predicates	10.553	1535	1537	40.482	4977	4799
% Sentences without predicates	15.87%	23.33%	26.00%	0%	0%	0%
% Sentences with predicates	84.13%	76.67%	74.00%	100%	100%	100%
# Unique predicates	3107	1074	1025	3107	1074	1025
# Unique arguments	57	43	42	57	43	42
Average # of tokens per sentence	16.3	12.6	12.1	25.40	21.11	21.08

Table 3: Corpus Statistics before and after preprocessing.

5 Feature-based Experiments

5.1 Model Description

Logistic Regression is a discriminative classifier often used in supervised machine learning algorithms for classification tasks. For binary classification problems, where the outcome is typically encoded as 0 or 1, it is particularly useful for situations where the relationship between the independent variables and the dependent binary variable is not linear but can be described by the logistic function.

According to Jurafsky and Manning (2012), Logistic Regression estimates the probabilities using a logistic function, which is an S-shaped curve that can take any real-valued number and map it into a value between 0 and 1, but never exactly at those limits see Figure 1. This characteristic makes it suitable for estimating probabilities.

Logistic Regression can be extended to handle multi-class classification through techniques such multinomial logistic regression. In the context of multi-class classification, logistic regression models the probabilities of different possible outcomes of a categorical dependent variable given a set of independent variables.

According to Jurafsky and Manning (2012), multinomial logistic regression replaces the logistic regression function used in binary logistic regression by the softmax function, which can handle multiple

[2]Example sentence: "Great atmosphere, great food."

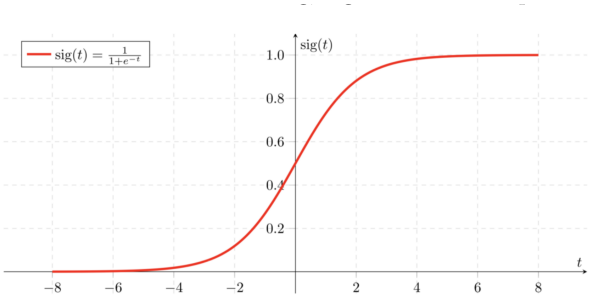


Figure 1: S-curved graph of logistic regression, taken from (TowardAI, 2021)

classes. This function computes the probability that a given instance belongs to each class, ensuring that the sum of these probabilities equals one. The predicted class is then the one with the highest probability.

Jurafsky and Manning (2012) also point out that while logistic regression is straightforward and efficient for binary classification, its extension to multi-class problems can introduce complexities, especially in terms of interpretation and computational requirements.

5.2 Selected Features

	Feature	Description
1.	Lemma (-1/+1)	Previous, current, and the next lemma
2.	POS (-1/+1)	Universal POS tags of previous, current, and next token
3.	Dependency relation	Dependency label of the token
4.	Dependency head	Dependency head token for the token
5.	Dependency path	Dependency path leading from token to predicate
6.	Dependency distance	Number steps in the dependency path from token to predicate
7.	Voice	Binary feature indicating active or passive voice in the sentence
8.	Named Entity type	Named Entity types with BIO tags per each token
9.	Predicate	The single predicate of the sentence
10.	Distance to predicate	Physical distance between the token and predicate
11.	Position	Relative position of the token to the predicate: left or right or 0 for predicate
12.	Predicate arguments (Propbank)	Possible arguments that predicate can take according to Propbank entry *(e.g. ARG0, ARG1. etc.)*
13.	Predicate roles (Propbank)	Possible roles that a predicate can take according to Propbank entry *(e.g. 'nominate.01': ['nominator', 'candidate'])*

Table 4: Feature Set for Semantic Role Labeling

a) Lexical features

Lemma embeddings

Word embeddings are a type of word representation that captures the semantic, syntactic, and contextual nuances of words in a dense vector format. They are particularly useful in Semantic Role Labeling (SRL) tasks, where assigning accurate roles to words based on their context is important. Furthermore, embeddings can also help overcome challenges such as word ambiguity and polysemy, leading to better accuracy in SRL systems (Peters et al., 2018). This project attempted to apply the lemma embeddings to improve the system's performance. However, the vectors became too lengthy and complex for the Logistic Regression model and, due to computational constraints, we decided to rely on one-hot encoded features instead.

POS window

POS tags help in distinguishing between different grammatical categories such as nouns, verbs, adjectives, and adverbs, which can significantly influence role assignment. The concept of "POS windows" refers to the use of POS tags from the context surrounding a token. Hacioglu (2004) suggests

that incorporating the POS tags of words surrounding the predicate can provide valuable syntactic cues that help in identifying the semantic roles of those words. For example, verbs typically act as predicates, while nouns are more likely to serve as subjects or objects, and by looking at the direct context of those tokens the feature helps the model to grasp the grammatical patterns that tend to surround different roles. Furthermore, Màrquez et al. (2005) emphasize the combination of these syntactic features with other lexical and semantic information to enhance the overall performance of the SRL system.

b) Dependency-based features

Dependency-based features have been established to be essential for SRL tasks as they provide structural and contextual information that helps identify semantic roles (Palmer et al., 2010). Below this section will describe the dependency features used in this project and why they are beneficial for SRL tasks according to Palmer et al. (2010).

Dependency relation of the token

In SRL, identifying the dependency relation (such as subject, object, or modifier) can help in determining the relationship between different tokens with the main predicate. For instance, subjects and objects are often likely candidates for certain semantic roles like agent and patient, respectively. Understanding these relations aids the SRL model in disambiguating role assignments, especially in complex sentences where multiple entities and actions are involved (Palmer et al., 2010).

Dependency head of the token

The dependency head of the token gives information about the hierarchical structure of a sentence and the token's function and relation to its head. This feature can be very beneficial for SRL because it shows the immediate connection between predicates and other tokens. For instance, tokens directly connected to the verb as direct objects are more likely to be recipients or targets in the action described (Palmer et al., 2010).

Dependency path from token to the predicate

The dependency path consists of a series of tokens and the dependency relations which connects a token to the predicate in a dependency tree. Analyzing the dependency path helps in understanding the syntactic construction of the sentence and how different elements contribute to the meaning related to the predicate. This feature is particularly useful for identifying indirect relationships and complex syntactic constructions that might influence the semantic role of a token. For instance, tokens involved in nested or subordinate clauses may have longer and more complex dependency paths to the main predicate, which could affect their semantic roles (Palmer et al., 2010).

Dependency distance between token and the predicate

The dependency distance between a token and the predicate measures the number of edges between them in the dependency tree. This feature is beneficial for SRL as it provides insights into the closeness of the relationship between the token and the predicate. Generally, tokens with shorter dependency distances to the predicate are more likely to have primary roles (such as agent or patient), while those farther away might have secondary roles (such as time or location) (Palmer et al., 2010).

c) Semantic features

Predicate

Most participants of the CoNLL-2005 Shared Task utilised some kind of information about the predicate itself. The predicate feature provides not only lexical information for the model, but also insight about the predicate's role set (Carreras and Màrquez, 2005). This is especially important in case of polysemous verbs, such as the verb "break", where different role sets of the verb trigger different number of and type of arguments, as shown in PropBank.[3]

[3] See: https://propbank.github.io/v3.4.0/frames/alias-break.html#break

Named Entity type
Named Entity types are important indicators of semantic roles as it is likely for certain named entity sequences in sentences to correspond to specific semantic role patterns. For example, "*[PERSON]* donated money to the *[ORGANIZATION]* in *[DATE]*". Named Entity types are shown to be useful in recognizing semantic roles (Sequeira et al., 2012).

Predicate arguments and roles
Specifying predicate arguments and roles from the PropBank corpus ensures model consistency within a standardized semantic framework. However, since the corpus provided by NLTK was not up-to-date, some entries were missing and could not be retrieved. For these predicates, we represented the argument and role feature as an empty list. Unfortunately, due to the large vector space, we could not implement this feature with the other ones since it caused memory issues.

d) Contextual features
Distance to predicate
Carreras and Màrquez (2005) describe that most participants in the CoNLL-2005 Shared Task extracted features measuring relative distances between sentence parts. Computing the distance between the token and predicate is useful due to the proximity bias, meaning that certain argument types, such as ARG0 and ARG1, tend to be closer the predicate than others, especially adjuncts.

Voice
Based on the work of Gildea and Jurafsky (2002), the voice feature was implemented. Providing the model information about the active or passive nature of verbs in the sentence can help the model distinguish between semantic roles and thereby reduce ambiguity. In English, the active voice typically follows a Subject-Verb-Object order, whereas the passive voice tends to evoke a Subject-Auxiliary-Verb-Object order. With the help of this feature, the model is more likely to classify "the door" as ARG1 and "John" as ARG0 in the passive sentence "The door was opened by John".

Position
The authors also implemented the position feature, which marks if the token is before or after the predicate in the sentence. This feature can be useful for assigning the appropriate semantic role, since agent-like arguments tend to appear before the predicate and patient- or theme-like arguments often come after the predicate in active sentences (Gildea and Jurafsky, 2002).

6 Results

The results in Table 5 display the performance of the Logistic Regression model on predicting the labels in the test data. Overall, the macro average across classes is 0.42 for F1-score, 0.48 for precision and 0.42 for recall. As anticipated, the model is best at predicting non-argument tokens, which take up about 91% of the total number of 101.144 tokens. ARG1, ARG0 and ARG2 are the most frequent argument labels, for each label the model achieved an F1-score higher than 0.74, with fairly balanced precision and recall scores. Concerning the other argument labels, the model performs worse for predicting ARG3, ARG4 and ARG5, which might be due to the rare occurrence of these labels. Moreover, the model reaches an F1-score above 0.55 for referential (R-) and continuous (C-) arguments if the label's occurrence is higher than 50. This also links back to the relatively low frequency of these labels across the entire dataset.

The classifier's performance on adjuncts presents a mixed outcome. For instance, negations (ARGM-NEG) and modals (ARGM-MOD) achieve F1 scores above .90, a notable achievement given their high sample sizes. Temporal (ARGM-TMP) and adjectival (ARGM-ADJ) adjuncts also perform well, each securing an .80 F1 score with relatively high amount of examples within the group. The extent adjuncts (ARGM-EXT), despite having fewer examples, demonstrate noticably high performance at .75 in F1 score.

Conversely, locatives (ARGM-LOC) and adverbials (ARGM-ADV) are underperforming despite a relatively high number of examples within the adjunct group. This suggests difficulty in accurately classifying these specific modifier types. The reference modifiers (R-), except for R-ARGM-LOC, all fail to score, indicating a complete lack of correct predictions. Similarly, discontinuous (C-) modifiers also register zero scores, further highlighting specific areas where the classifier struggles.

The presented findings face several limitations. The model is expected to perform differently depending on the selected features, the domain of the test data, the label distribution and the model's parameters. First, the imbalanced distribution of labels significantly impacts the classifier's ability to learn infrequent categories effectively, in future work data augmentation techniques could address this issue. Second, further experimentation with possible feature combinations as part of a feature ablation study could improve the model's performance. Third, the static nature of the feature-based model is not as context-sensitive and dynamic as neural networks that might handle the task better across different domains and genres.

7 Conclusions

The task involved identifying and classifying arguments and adjuncts in Semantic Role Labeling, using the Universal Proposition Bank. Key features included dependency-based, semantic, and contextual features, among others. The Logistic Regression model achieved high performance for non-argument tokens and varied success across different argument types, with the highest frequencies among ARG1, ARG0, and ARG2. The classification of modifiers shows mixed results, with difficulty in accurately classifying locatives, adverbials, and almost all reference modifiers while performing well at temporal modifiers, negations, and modals. One final aspect to consider in the performance of SRL models is the impact of feature selection and data balance. This highlights areas for future improvement and research in advanced NLP applications.

Label	Precision	Recall	F1-Score	Support
O	0.99	0.99	0.99	91725
ARG0	0.78	0.77	0.77	1733
ARG1	0.81	0.81	0.81	3241
ARG2	0.74	0.75	0.74	1129
ARG3	0.58	0.30	0.39	74
ARG4	0.59	0.61	0.60	56
ARG5	0.00	0.00	0.00	1
R-ARG0	0.74	0.84	0.78	67
R-ARG1	0.59	0.62	0.60	52
R-ARG2	0.00	0.00	0.00	1
C-ARG0	0.00	0.00	0.00	3
C-ARG1	0.84	0.40	0.55	52
C-ARG1-DSP	0.00	0.00	0.00	1
C-ARG2	0.67	0.57	0.62	7
C-ARG3	0.00	0.00	0.00	2
ARG1-DSP	0.00	0.00	0.00	4
ARGA	0.00	0.00	0.00	2
ARGM-TMP	0.82	0.78	0.80	543
ARGM-ADV	0.61	0.58	0.59	496
ARGM-MOD	0.94	0.95	0.94	442
ARGM-ADJ	0.76	0.82	0.79	228
ARGM-NEG	0.90	0.96	0.93	216
ARGM-LOC	0.58	0.52	0.55	207
ARGM-DIS	0.77	0.65	0.71	182
ARGM-MNR	0.58	0.30	0.39	148
ARGM-EXT	0.76	0.74	0.75	105
ARGM-PRP	0.52	0.43	0.47	75
ARGM-LVB	0.67	0.65	0.66	69
ARGM-PRR	0.68	0.41	0.51	69
ARGM-CAU	0.39	0.24	0.30	46
ARGM-DIR	0.43	0.32	0.37	47
ARGM-PRD	0.59	0.23	0.33	44
ARGM-GOL	1.00	0.04	0.08	24
ARGM-COM	0.75	0.46	0.57	13
ARGM-CXN	0.55	0.50	0.52	12
R-ARGM-LOC	0.86	0.67	0.75	9
R-ARGM-MNR	0.00	0.00	0.00	8
R-ARGM-TMP	0.00	0.00	0.00	2
R-ARGM-ADJ	0.00	0.00	0.00	1
R-ARGM-ADV	0.00	0.00	0.00	1
R-ARGM-DIR	0.00	0.00	0.00	1
C-ARGM-CXN	0.00	0.00	0.00	5
C-ARGM-LOC	0.00	0.00	0.00	1
accuracy				0.97
macro avg	0.48	0.39	0.42	101144
weighted avg	0.97	0.97	0.97	101144

Table 5: Comprehensive Classification Report

References

Claire Bonial, Jena Hwang, Julia Bonn, Kathryn Conger, Olga Babko-Malaya, and Martha Palmer. 2012. English propbank annotation guidelines. *Center for Computational Language and Education Research Institute of Cognitive Science University of Colorado at Boulder*, 48.

Xavier Carreras and Lluís Màrquez. 2005. Introduction to the CoNLL-2005 shared task: Semantic role labeling. In Ido Dagan and Daniel Gildea, editors, *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, pages 152–164, Ann Arbor, Michigan. Association for Computational Linguistics.

Andrea Gesmundo, James Henderson, Paola Merlo, and Ivan Titov. 2009. A latent variable model of synchronous syntactic-semantic parsing for multiple languages. In *CoNLL Shared Task*.

Daniel Gildea and Dan Jurafsky. 2000. Automatic labeling of semantic roles. In *Annual Meeting of the Association for Computational Linguistics*.

Daniel Gildea and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288.

Kadri Hacioglu. 2004. Semantic role labeling using dependency trees. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 1273–1276.

Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, Pavel Straňák, Mihai Surdeanu, Nianwen Xue, and Yi Zhang. 2009. The CoNLL-2009 shared task: Syntactic and semantic dependencies in multiple languages. In Jan Hajič, editor, *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task*, pages 1–18, Boulder, Colorado, June. Association for Computational Linguistics.

James Henderson, Paola Merlo, Ivan Titov, and Gabriele Musillo. 2013. Multilingual joint parsing of syntactic and semantic dependencies with a latent variable model. *Computational Linguistics*, 39(4):949–998, December.

Daniel Jurafsky and Christopher D. Manning. 2012. *Speech and Language Processing*. Prentice Hall, Upper Saddle River, New Jersey, USA, 2 edition.

Diego Marcheggiani, Anton Frolov, and Ivan Titov. 2017. A simple and accurate syntax-agnostic neural model for dependency-based semantic role labeling. *ArXiv*, abs/1701.02593.

Lluís Màrquez, Pere Comas, Jesús Giménez, and Neus Català. 2005. Semantic role labeling as sequential tagging. In Ido Dagan and Daniel Gildea, editors, *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, pages 193–196, Ann Arbor, Michigan, June. Association for Computational Linguistics.

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

Martha Palmer, Daniel Gildea, and Nianwen Xue. 2010. Semantic role labeling. *Synthesis Lectures on Human Language Technologies*, 3(1):1–103.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 2227–2237.

Sameer Pradhan, Kadri Hacioglu, Wayne H. Ward, James H. Martin, and Dan Jurafsky. 2005. Semantic role chunking combining complementary syntactic views. In *Conference on Computational Natural Language Learning*.

Vasin Punyakanok, Dan Roth, and Wen-tau Yih. 2008. The importance of syntactic parsing and inference in semantic role labeling. *Computational Linguistics*, 34(2):257–287.

Michael Roth and Mirella Lapata. 2016. Neural semantic role labeling with dependency path embeddings. *ArXiv*, abs/1605.07515.

J. Sequeira, T. Gonçalves, and P. Quaresma. 2012. Semantic role labeling for portuguese – a preliminary approach –. In *Computational Processing of the Portuguese Language. PROPOR 2012. Lecture Notes in Computer Science*, volume 7243. Springer, Berlin, Heidelberg.

Tianze Shi, Igor Malioutov, and Ozan Irsoy. 2020. Semantic role labeling as syntactic dependency parsing. *ArXiv*, abs/2010.11170.

Mihai Surdeanu, Richard Johansson, Adam Meyers, Lluís Màrquez i Villodre, and Joakim Nivre. 2008. The conll 2008 shared task on joint parsing of syntactic and semantic dependencies. In *Conference on Computational Natural Language Learning*.

TowardAI. 2021. Sentiment analysis with logistic regression.

Yuhao Zhang, Peng Qi, and Christopher D. Manning. 2018. Graph convolution over pruned dependency trees improves relation extraction. *ArXiv*, abs/1809.10185.

Appendix 1: distribution of work

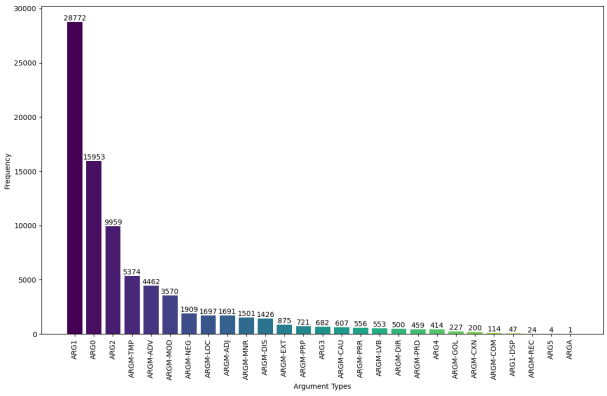
Selin Acikel Writing sections: Introduction, 5.1 Model description, 5.2 Selected features: dependency-based features. Writing code for the extraction of dependency features.

Murat Ertas Writing Related work, part of results, part of semantic feature descriptions, NER and Propbank feature extraction scripts.

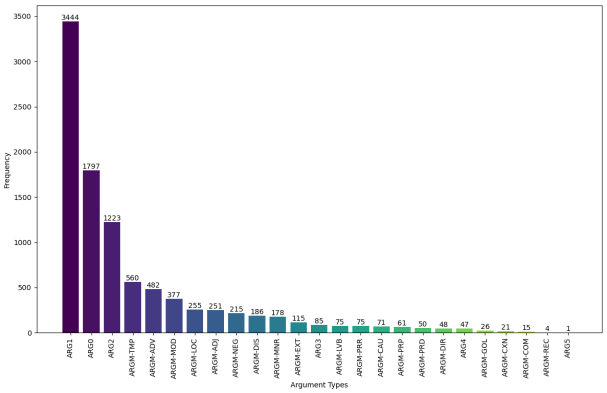
Tessel Haagen Code for reading data in, semantic features, improving feature extracting code, merging features, and smooth pipeline. Wrote lexical features in selected features and conclusion.

Csenge Szabó Writing sections: Task Description, Pipeline Description, Dataset Description, Preprocessing, part of Selected Features and Results. Code for extracting contextual features, corpus statistics, creating and evaluating the model.

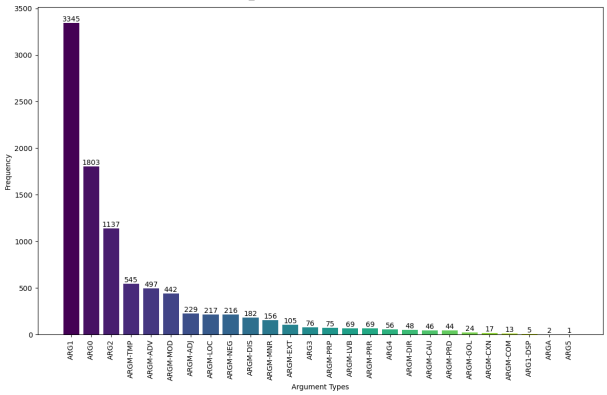
Appendix 2



(a) Training set distribution



(b) Development set distribution



(c) Test set distribution

Figure 2: Argument-label Distribution in the Dataset