Problem Statement

A number of pairs of homonyms have been selected from the Russian online dictionary of homonyms. The project goal is to apply supervised learning methods to analyze the accuracy of the achieved results in disambiguation of homonymous pairs. In contrast of previous work, a focus here was put on building bigger training and evaluation sets for fewer pairs of homonyms to create a more accurate training model and perform a more accurate evaluation of the results.

The homonyms analyzed were either lexicogrammatical (different parts of speech and semantic differences), or morphosyntactic (different parts of speech, semantic and phonetic differences). The typology of the homonyms analyzed can be defined in degrees, ranging from partial to full homonyms (that remain identical in all of their word-forms). Only partial homonyms were analyzed here.

Word	**Grammatical categories**	**Translation**	**Transcription**
Пила	***Noun****: feminine inanimate singular nominative first declension*	A saw (tool)	[pʲɪˈɬa]
Пила	***Verb****: feminine singular past indicative imperfective of “пить”*	Drank	[pʲɪˈɬa]
Дуло	***Noun****: neutral inanimate singular nominative/accusative second declension*	A barrel/ muzzle (of a firearm)	[ˈdulə]
Дуло	***Verb****: neutral singular past indicative imperfective of “дуть”*	Was drafty/ blew (wind)	[ˈdulə]
Жаркое	***Noun****: neuter inanimate singular nominative/accusative second declension*	A stew/ roast (meat)/ stir-fry	[ʐɐrˈkojə]
Жаркое	***Adjective****: neuter long singular nominative/accusative of “жаркий”*	Hot/ tropical/ fervent	[ˈʐarkəjə]
Печь	***Noun****: feminine inanimate singular nominative/accusative third declension*	A stove/ oven/ furnace	[pʲet͡ɕ]
Печь	***Verb****: infinitive imperfective*	To bake/ scorch	[pʲet͡ɕ]

The overall goal is to create a disambiguation engine which accepts a sentence and a homonym to disambiguate and returns the most likely disambiguation of that homonym. Another goal is to analyse various combinations of features to determine the most favorable feature-to-data ratio and how it affects data training, what information is being extracted from each set of features, and where the overlap of identical training information occurs.

Solution Approach

I employed a supervised machine learning approach to the problem. Given the list of homonyms to disambiguate, I created a list of the possible parts of speech associated to each homonym using a Russian morphological analyzer. I used this to create labeled training and test data sets of correctly disambiguated homonyms with the following structure:

word	word_id	sent	start	stop
жаркое	жаркое_noun	*Такое жаркое с черносливом готовит только моя мама.*	7	13
жаркое	жаркое_adjf	*Климат резко континентальный: холодная зима и жаркое лето.*	47	53

The correct word id was assigned to each sentence by myself or from the pre-disambiguated section of the Russian National Corpus (RNC).

The pymorphy2 and nltk packages were used throughout this work for tokenization and lemmatization, and in order to assess which possible parts of speech could apply to a homonym using the morphological analyzer in the same package.

Scikit-learn was used to build a machine learning classifier using a linear support vector classifier optimized by the stochastic gradient descent. Three types of sentence vectorizers were then used to create features (HashingVectorizer, CountVectorizer, TfidfVectorizer).

The linear SVM showed the highest level of accuracy in comparison to MLP or Random Forest classifiers. The knowledge sources used were parts of speech of neighboring words (both in narrow and global context), distance (number of words in between) to

each of the closest parts of speech, single words in the surrounding context and lemmas or “normal forms” of the words in the surrounding context.

Three steps of the feature creation were as follows:

- TfIdfVectorizer was used on the entire corpus of surrounding words
- HashingVectorizer was used on the parts of speech in the global context
- CountVectorizer was used on the parts of speech in the local context as well as word lemmas in a more narrow context

N-gram ranges were tried as well, but did not produce any significant improvement to the score due to the unfavorable feature-to-data length ratio.

The following sets of features and their combinations were tested:

- single words in the surrounding context
- parts of speech in the global context (100 tokens)
- parts of speech in the local context (2-7 tokens)
- word lemmas in a more narrow context (20 tokens)
- distance to the nearest parts of speech going backwards
- distance to the nearest parts of speech going forward

When determining the relative importance of each type of feature, no significant differences were found. However, the most favorable sets of features for the words tested was as follows (given the current feature-to-data ratio):

1. Parts of speech in the global context & word lemmas in a more narrow context:

- “Dulo”: 91% / 54% baseline
- “Pila”: 83% / 60% baseline
- “Pech”: 86% / 78% baseline
- “Zharkoe”: 86% / 69% baseline

2. Parts of speech in the global context & single words in the surrounding context

3. Parts of speech in the global context & word lemmas in a more narrow context

Accuracy was calculated relative to the baseline. The baseline was defined as the frequency of the most frequently occurring class in the training data. The meaning for

the baseline is the score of the trivial classifier which constantly predicts the most frequent class for all feature vectors.

Details of Method

In order to disambiguate a given sentence, such as:

> *Спереди торчало **дуло**, а вдоль корпуса, как сложенные за спиной руки, лежали оба манипулятора.*

The following steps were employed:

1. Train and test datasets were concatenated to apply a random split later for cross-validation. Starting with one complete labeled dataset of sentences where each sentence has at least one instance of the target current homonym.
2. The full dataset was tokenized and the ambiguous homonym isolated.
3. The Russian morphological analyzer was used to create a corpus of possible parts of speech (PoS corpus) for all other words in the tokenized corpus.
4. The PoS corpus was used to isolate a certain number of preceding and succeeding words' parts of speech (both the closest neighbors and global context neighbors).
5. The Russian morphological analyzer was then used again to create a corpus of all tokens' lemmas in order to analyze/ include a number of neighboring words' lemmas as features.
6. CountVectorizer, HashingVectorizer and TfidfVectorizer were used to create features and feed them into the classifier.
7. All feature columns were concatenated into a single dataframe, then this dataframe was split into the train and the test data set on which the classifier was trained and tested respectively, and split using the process equivalent to the shuffle split (at random places with each split being of the same size where the splits can overlap).
8. The mean and standard deviation of the classifier scores across the validation splits were computed.
9. The confusion matrix for each train/test split score cycle was generated for the PoS predictions and subsequently written into a dataframe.

Results

The sample table below shows which predictions were correct and which were not. The target word is highlighted in red:

	sent	ground_truth	preds	success
0	где-нибудь на свадьбе мог после официальной части подойти к буфету и взять жаркое руками	0	0	True
1	кусок сухого бульона распустить в 1/ 2 стакана воды, снять с соуса из-под жаркого жир, смешать с распущенным сухим бульоном; прибавить капорцов или лука-шарлот, поджаренного в жире, собранном с соуса, или маринованных боровиков или рыжиков, вскипятить, облить сложенное на блюдо жаркое .	0	0	True
2	живёт он с женой в хорошем деревенском доме, с верандой для отдыха, увитой виноградом; при доме есть и хозяйство. время жаркое — полдень. на открытой веранде подполковник гуров и его гость алибеков; разморённые обедом, они дремлют в лёгких плетёных креслах в ожидании чая.	1	0	False
3	вода тёплая и маслянистая. слишком жаркое лето. даже в ста метрах от берега — тёплая.	1	1	True
4	проходила весна, начиналось лето. лето жаркое , с частыми грозами. ветром трепало ивы на улице.	1	1	True
5	такое жаркое с черносливом готовит только моя мама.	0	0	True

The overall results accuracy for the given words was approximately in the range of 83-90%:

Zharkoe	(0.8696969696969697, ' +/-', 0.0018457300275482075) 0.7272727272727273 1.1958333333333333	[[8 1] [2 22]]
Dulo	(0.9199999999999999, ' +/-', 0.0007199999999999991) 0.56 1.6428571428571426	[[19 3] [2 26]]

Pila	(0.8300000000000001, ' +/-', 0.0022250000000000004) 0.525 1.580952380952381	[[15 4] [0 21]]
Pech	(0.8594594594594595, ' +/-', 0.00537618699780862) 0.8108108108108109 1.0599999999999998	[[28 2] [3 4]]

Future Goals

1. Make use of Sphinx for documentation (including exceptions and logging)
2. Experiment with different tokenizers better customized for Russian, and apply them to a bigger dataset (spacey_russian_tokenizer, razdel)
3. Perform a more in-depth feature evaluation and feature engineering; develop feature reduction strategies
4. Perform a more in-depth analysis of the systematic patterns in the misclassified homonyms and assess how these errors can be corrected
5. Include more labeled data as well as more homonymous pairs of other parts of speech
6. Experiment with DictVectorizer and encoding word position into the features for the closest neighbor words to the disambiguation target
7. Add a development set
8. Incorporate unlabeled data using clustering algorithms to automatically generate classes.

Attachments:

1. Python code: disamb.py
2. Python code: test.py
3. Training data set (“dulo”, “pech”, “zharkoe”, “pila”)
4. Test data set (“dulo”, “pech”, “zharkoe”, “pila”)
5. Results table for each word (html)
6. Project description