# Using Context Features for Morphological Analysis of Russian

Alexey Sorokin[1,2], Ekaterina Yankovskaya[1]

[1]Moscow State University, [2]Moscow Institute of Science and Technology

“Dialogue”, International Conference
on Computational Linguistics,
Moscow, June, 1st, 2017

# POS-tagging for Russian and English

- POS-tagging for English
  - Plenty of systems and approaches: HMM, CRF, dependency networks, neural networks, combinations of approaches...
  - High results due to relatively simple morphology ($\approx$ 97.5% on WSJ).

# POS-tagging for Russian and English

- POS-tagging for English
  - Plenty of systems and approaches: HMM, CRF, dependency networks, neural networks, combinations of approaches...
  - High results due to relatively simple morphology ($\approx$ 97.5% on WSJ).
- POS-tagging for Russian: problems with traditional approaches

  - HMM do not decompose tags and uses only 2 previous words. Though simple to implement and fast.

## POS-tagging for Russian and English

- POS-tagging for English
  - Plenty of systems and approaches: HMM, CRF, dependency networks, neural networks, combinations of approaches...
  - High results due to relatively simple morphology ($\approx$ 97.5% on WSJ).
- POS-tagging for Russian: problems with traditional approaches
  - HMM do not decompose tags and uses only 2 previous words. Though simple to implement and fast.
  - CRF do decompose tags but creates too much features. History of length 2 is already problematic to handle.
- Even if neural networks work well we do not know why. Let's do some linguistics instead.

# POS-tagging for Russian and English

- POS-tagging for English
  - Plenty of systems and approaches: HMM, CRF, dependency networks, neural networks, combinations of approaches…
  - High results due to relatively simple morphology ($\approx$ 97.5% on WSJ).
- POS-tagging for Russian: problems with traditional approaches
  - HMM do not decompose tags and uses only 2 previous words. Though simple to implement and fast.
  - CRF do decompose tags but creates too much features. History of length 2 is already problematic to handle.
  - And in Russian we need history of arbitrary length.
- Even if neural networks work well we do not know why. Let's do some linguistics instead.

# POS-tagging for Russian and English

- POS-tagging for English
  - Plenty of systems and approaches: HMM, CRF, dependency networks, neural networks, combinations of approaches...
  - High results due to relatively simple morphology ($\approx$ 97.5% on WSJ).
- POS-tagging for Russian: problems with traditional approaches
  - HMM do not decompose tags and uses only 2 previous words. Though simple to implement and fast.
  - CRF do decompose tags but creates too much features. History of length 2 is already problematic to handle.
  - And in Russian we need history of arbitrary length.
  - Constraint-based approach do not handle complex cases or require too much labour.
- Even if neural networks work well we do not know why. Let's do some linguistics instead.

# POS-tagging for Russian and English

- POS-tagging for English
  - Plenty of systems and approaches: HMM, CRF, dependency networks, neural networks, combinations of approaches...
  - High results due to relatively simple morphology ($\approx$ 97.5% on WSJ).
- POS-tagging for Russian: problems with traditional approaches
  - HMM do not decompose tags and uses only 2 previous words. Though simple to implement and fast.
  - CRF do decompose tags but creates too much features. History of length 2 is already problematic to handle.
  - And in Russian we need history of arbitrary length.
  - Constraint-based approach do not handle complex cases or require too much labour.
  - Neural networks.. Hmm, they were not tested.
- Even if neural networks work well we do not know why. Let's do some linguistics instead.

# Linguistics for computational morphology

- Common ambiguities in Russian:
  - Nominative vs accusative for nouns and adjectives.
  - Genitive vs accusative for nouns and adjectives.

## Linguistics for computational morphology

- Common ambiguities in Russian:
  - Nominative vs accusative for nouns and adjectives.
  - Genitive vs accusative for nouns and adjectives.
  - Short adjectives vs adverbs.
  - “что” — a pronoun or a conjunction?

## Linguistics for computational morphology

- Common ambiguities in Russian:
    - Nominative vs accusative for nouns and adjectives.
    - Genitive vs accusative for nouns and adjectives.
    - Short adjectives vs adverbs.
    - “что” — a pronoun or a conjunction?
- How we may process it:
    - A nominative is usually a subject.
    - Accusative often follows a transitive verb being its direct object.
    - Adjectives and nouns agree in case, gender and number.

# Linguistics for computational morphology

- Common ambiguities in Russian:
  - Nominative vs accusative for nouns and adjectives.
  - Genitive vs accusative for nouns and adjectives.
  - Short adjectives vs adverbs.
  - “что” — a pronoun or a conjunction?
- How we may process it:
  - A nominative is usually a subject.
  - Accusative often follows a transitive verb being its direct object.
  - Adjectives and nouns agree in case, gender and number.
  - Short adjective is usually a predicate etc.

# Linguistics for computational morphology

- Common ambiguities in Russian:
  - Nominative vs accusative for nouns and adjectives.
  - Genitive vs accusative for nouns and adjectives.
  - Short adjectives vs adverbs.
  - “что” — a pronoun or a conjunction?
- How we may process it:
  - A nominative is usually a subject.
  - Accusative often follows a transitive verb being its direct object.
  - Adjectives and nouns agree in case, gender and number.
  - Short adjective is usually a predicate etc.
- Let’s extract features reflecting whether these constraints are satisfied.

# Linguistics for computational morphology

- Common ambiguities in Russian:
  - Nominative vs accusative for nouns and adjectives.
  - Genitive vs accusative for nouns and adjectives.
  - Short adjectives vs adverbs.
  - "что" — a pronoun or a conjunction?
- How we may process it:
  - A nominative is usually a subject.
  - Accusative often follows a transitive verb being its direct object.
  - Adjectives and nouns agree in case, gender and number.
  - Short adjective is usually a predicate etc.
- Let's extract features reflecting whether these constraints are satisfied.
- These features are "soft constraints".

# Soft constraints

- Hard constraint: a full adjective must be coordinated with some noun. These two words agree in case, gender and number.
- Hard constraint: a transitive verb must be followed or preceded by a direct object.

# Soft constraints

- Hard constraint: a full adjective must be coordinated with some noun. These two words agree in case, gender and number.
- Hard constraint: a transitive verb must be followed or preceded by a direct object.
- Hard constraint often fail:
  - *Рассказал сказку* vs *рассказал друзьям о себе*.
  - *Думал уйти* vs *Думал о погоде*.

# Soft constraints

- Hard constraint: a full adjective must be coordinated with some noun. These two words agree in case, gender and number.
- Hard constraint: a transitive verb must be followed or preceded by a direct object.
- Hard constraint often fail:
  - *Рассказал сказку* vs *рассказал друзьям о себе*.
  - *Думал уйти* vs *Думал о погоде*.
- Soft constraint: let us count a number of transitive verbs followed by a direct object.

# Soft constraints

- Hard constraint: a full adjective must be coordinated with some noun. These two words agree in case, gender and number.
- Hard constraint: a transitive verb must be followed or preceded by a direct object.
- Hard constraint often fail:
    - *Рассказал сказку* vs *рассказал друзьям о себе*.
    - *Думал уйти* vs *Думал о погоде*.
- Soft constraint: let us count a number of transitive verbs followed by a direct object.
- That would be a strong positive feature.

## Feature inventory

9 groups of features:

- Adjective coordination.
- Determiner cooordination.
- Preposition government.

# Feature inventory

9 groups of features:

- Adjective coordination.
- Determiner cooordination.
- Preposition government.
- Verb government.
- Nominative features.
- Accusative features.

# Feature inventory

9 groups of features:

- Adjective coordination.
- Determiner cooordination.
- Preposition government.
- Verb government.
- Nominative features.
- Accusative features.
- Noun-noun features.
- Noun-and-noun features.
- Noun-comma-noun features.

# Examples of features: adjectives.

- Adjectives:
  - Number of adjectives.
  - Number of adjectives, coordinated with nouns to the right side.
  - Number of adjectives, coordinated with nouns to the left side.
  - Indicator for non-coordinated adjectives presence.

# Examples of features: adjectives.

- Adjectives:
  - Number of adjectives.
  - Number of adjectives, coordinated with nouns to the right side.
  - Number of adjectives, coordinated with nouns to the left side.
  - Indicator for non-coordinated adjectives presence.
- Determiners: the same as adjectives.

# Examples of features: adjectives.

- Adjectives:
  - Number of adjectives.
  - Number of adjectives, coordinated with nouns to the right side.
  - Number of adjectives, coordinated with nouns to the left side.
  - Indicator for non-coordinated adjectives presence.
- Determiners: the same as adjectives.
- Prepositions:
  - Number of prepositions.
  - Number of prepositions, coordinated with nouns in case.
  - Indicator of non-coordinated prepositions presence.

# Examples of features: verb government

- For every verb lemma we collect the counts of following noun group cases.
- For every verb lemma we collect the counts of following preposition group cases.

## Examples of features: verb government

- For every verb lemma we collect the counts of following noun group cases.
- For every verb lemma we collect the counts of following preposition group cases.
- Extracted features:
    - Sum of log-probabilities of verb objects over all verbs in the sentence.
    - Sum of log-probabilities of preposition verb objects over all verbs in the sentence.

# Examples of features: verb government

- For every verb lemma we collect the counts of following noun group cases.
- For every verb lemma we collect the counts of following preposition group cases.
- Extracted features:
  - Sum of log-probabilities of verb objects over all verbs in the sentence.
  - Sum of log-probabilities of preposition verb objects over all verbs in the sentence.
  - Number of reflexive verbs followed by nominative (strong positive feature).
  - Number of reflexive verbs followed by instrumental case.
  - Total number of verbs in the sentence.

# Examples of features: nominatives

- Nominatives: about 20 features.
  - Number of nominatives coordinated with verbs to the right.
  - Number of nominatives coordinated with verbs to the left.

## Examples of features: nominatives

- Nominatives: about 20 features.
  - Number of nominatives coordinated with verbs to the right.
  - Number of nominatives coordinated with verbs to the left.
  - Number of nominative-nominative clauses.
  - Number of *это*-nominative clauses.
  - Number of noun-adjective clauses etc.

# Examples of features: nominatives

- Nominatives: about 20 features.
  - Number of nominatives coordinated with verbs to the right.
  - Number of nominatives coordinated with verbs to the left.
  - Number of nominative-nominative clauses.
  - Number of *это*-nominative clauses.
  - Number of noun-adjective clauses etc.
- Accusatives: about 20 features.
  - Number of transitive verbs.
  - Number of transitive verbs followed by accusative/genitive.
  - Number of transitive verbs preceded by *не* and followed by accusative/genitive.
  - Number of transitive verbs with direct objects to the left etc.

# The learning algorithm

- The main idea: train a linear classifier to rank correct hypotheses higher.

# The learning algorithm

- The main idea: train a linear classifier to rank correct hypotheses higher.
- Training procedure:
  - Generate *n*-best hypotheses for each sentence in the training set using the baseline classifier.
  - For each hypothesis extract a feature vector.

## The learning algorithm

- The main idea: train a linear classifier to rank correct hypotheses higher.
- Training procedure:
  - Generate $n$-best hypotheses for each sentence in the training set using the baseline classifier.
  - For each hypothesis extract a feature vector.
  - On each sentence $s_i$, train the classifier to score the feature vector $x_{i,0}$ higher than vectors $x_{i,j}$ for other hypotheses $s_j$:

$$(w; x_{i,0}) > (w; x_{i,j}).$$

# The learning algorithm

- The main idea: train a linear classifier to rank correct hypotheses higher.
- Training procedure:
  - Generate $n$-best hypotheses for each sentence in the training set using the baseline classifier.
  - For each hypothesis extract a feature vector.
  - On each sentence $s_i$, train the classifier to score the feature vector $x_{i,0}$ higher than vectors $x_{i,j}$ for other hypotheses $s_j$:

$$(w; x_{i,0}) > (w; x_{i,j}).$$

  - Equivalently,

$$(w; x_{i,0} - x_{i,j}) > 0.$$

# The learning algorithm

- The main idea: train a linear classifier to rank correct hypotheses higher.
- Training procedure:
  - Generate $n$-best hypotheses for each sentence in the training set using the baseline classifier.
  - For each hypothesis extract a feature vector.
  - On each sentence $s_i$, train the classifier to score the feature vector $x_{i,0}$ higher than vectors $x_{i,j}$ for other hypotheses $s_j$:

$$(w; x_{i,0}) > (w; x_{i,j}).$$

  - Equivalently,

$$(w; x_{i,0} - x_{i,j}) > 0.$$

  - Standard classification task: arrange $x_{i,0} - x_{i,j}$ to the positive class and the opposite vector to the negative one.

# The tagging algorithm

- The prediction procedure:
  - Generate *n*-best hypotheses for each sentence in the test set using baseline classifier.

# The tagging algorithm

- The prediction procedure:
  - Generate $n$-best hypotheses for each sentence in the test set using baseline classifier.
  - Using the trained vector $\mathbf{w}$ of weights, select the hypothesis $x_{i,j}$ with the highest score $(\mathbf{w}, x_{i,j})$.

# The tagging algorithm

- The prediction procedure:
  - Generate $n$-best hypotheses for each sentence in the test set using baseline classifier.
  - Using the trained vector $\mathbf{w}$ of weights, select the hypothesis $x_{i,j}$ with the highest score $(\mathbf{w}, x_{i,j})$.
- Algorithm: logistic regression. Averaged margin perceptron gives slightly worse results.

# Performance evaluation

| № | Model | Development set | | Test set | |
|---|---|---|---|---|---|
| | | Tag acc. | Sent acc. | Tag acc. | Sent acc. |
| 1 | HMM+prep+trans | 95.0 | 74.1 | 93.77 | 65.15 |
| 2 | 1+adj+det+prep | 95.3 | 74.3 | 94.05 | 66.14 |
| 3 | 2+verbs | 95.5 | 75.2 | 94.22 | 66.77 |
| 4 | 3+nom+acc | 96.2 | 78.1 | 94.75 | 68.79 |
| 5 | 4+conj+noun-noun | 96.3 | 78.5 | 94.82 | 69.32 |

Таблица: Results on development and test set of MorphoRuEval-2017

## Conclusions

- Positive:
  - Linguistic features and reranking actually work.

## Conclusions

- Positive:
  - Linguistic features and reranking actually work.
- Problems:
  - Careful and labour-intensive feature engeneering (otherwise only a marginal gain is achieved).
  - Basic classifier probability receives too much weight.

# Conclusions

- Positive:
  - Linguistic features and reranking actually work.
- Problems:
  - Careful and labour-intensive feature engeneering (otherwise only a marginal gain is achieved).
  - Basic classifier probability receives too much weight.
  - Reranking against lower hypotheses: basic classifier probability already does well.
  - Reranking against higher hypotheses: not all linguistic constraints are violated in such hypotheses.

# Future work

- Partial solutions:
  - Rerank only against hypotheses whose basic loss is lower than some threshold.
  - Subtract a margin from basic classifier gain (small positive gains become negative forcing the classifier to use other features).

# Future work

- Partial solutions:
  - Rerank only against hypotheses whose basic loss is lower than some threshold.
  - Subtract a margin from basic classifier gain (small positive gains become negative forcing the classifier to use other features).
- Future work:
  - Integrate a stronger basic classifier (CRF or neural nets).
  - Use more complex reranking procedure.

# Future work

- Partial solutions:
  - Rerank only against hypotheses whose basic loss is lower than some threshold.
  - Subtract a margin from basic classifier gain (small positive gains become negative forcing the classifier to use other features).
- Future work:
  - Integrate a stronger basic classifier (CRF or neural nets).
  - Use more complex reranking procedure.
  - Automatic feature selection from patterns.
  - Use more lexically-oriented features.

Спасибо за внимание!

Thank you for your attention!