Texts in, meaning out: neural language models in semantic similarity task for Russian

Andrey Kutuzov
Igor Andreev

Mail.ru Group
National Research University Higher School of Economics

28 May 2015
Dialogue, Moscow, Russia
1 Intro: why semantic similarity?
2 Going neural
3 Task description
4 Texts in: used corpora
5 Meaning out: evaluation
6 Twiddling the knobs: importance of settings
7 Russian National Corpus wins
8 What next?
9 Q and A
Intro: why semantic similarity?

- A means in itself: finding synonyms or near-synonyms for search query expansion or other needs.
Intro: why semantic similarity?

- A means in itself: finding synonyms or near-synonyms for search query expansion or other needs.
- Drawing a ‘semantic map’ of the language in question.
Intro: why semantic similarity?

- **A means in itself**: finding synonyms or near-synonyms for search query expansion or other needs.
- Drawing a ‘**semantic map**’ of the language in question.
- Convenient way to **estimate soundness of a semantic model** in general.
Intro: why semantic similarity?

TL;DR

1. Neural network language models (NNLMs) can be successfully used to solve semantic similarity problem for Russian.
2. Several models trained with different parameters on different corpora are evaluated and their performance reported.
3. Russian National Corpus proved to be one of the best training corpora for this task;
4. Models and results are available on-line.
Intro: why semantic similarity?

TL;DR

1. Neural network language models (NNLMs) can be successfully used to solve semantic similarity problem for Russian.
Intro: why semantic similarity?

TL;DR

1. Neural network language models (NNLMs) can be successfully used to solve semantic similarity problem for Russian.
2. Several models trained with different parameters on different corpora are evaluated and their performance reported.
Intro: why semantic similarity?

TL;DR

1. Neural network language models (NNLMs) can be successfully used to solve semantic similarity problem for Russian.
2. Several models trained with different parameters on different corpora are evaluated and their performance reported.
3. Russian National Corpus proved to be one of the best training corpora for this task;
Intro: why semantic similarity?

TL;DR

1. Neural network language models (NNLMs) can be successfully used to solve semantic similarity problem for Russian.

2. Several models trained with different parameters on different corpora are evaluated and their performance reported.

3. Russian National Corpus proved to be one of the best training corpora for this task;

4. Models and results are available on-line.
Intro: why semantic similarity?

So what is similarity?

Andrey Kutuzov  Igor Andreev (Mail.ru Group National Research University Higher School of Economics)

Texts in, meaning out: neural language models in semantic similarity task for Russian

28 May 2015 Dialogue, Moscow, Russia
Intro: why semantic similarity?

So what is similarity?
Intro: why semantic similarity?

So what is similarity?

And more: what is **lexical** similarity?
Intro: why semantic similarity?

How to represent meaning?

Semantics is difficult to represent formally. To be able to tell which words are semantically similar, means to invent machine-readable word representations with the following constraint: words which people feel to be similar should possess mathematically similar representations.

«Светильник» must be similar to «лампа» but not to «кипятильник», even though their surface form suggests the opposite.
Intro: why semantic similarity?

How to represent meaning?

- **Semantics** is difficult to represent formally.
Intro: why semantic similarity?

How to represent meaning?

- **Semantics** is difficult to represent formally.
- To be able to tell which words are **semantically similar**, means to invent machine-readable word **representations** with the following constraint: words which people feel to be similar should possess mathematically similar representations.
How to represent meaning?

- **Semantics** is difficult to represent formally.
- To be able to tell which words are **semantically similar**, means to invent machine-readable word **representations** with the following constraint: words which people feel to be similar should possess mathematically similar representations.
- «Светильник» must be similar to «лампа» but not to «кипятильник», even though their surface form suggests the opposite.
Intro: why semantic similarity?

Possible data sources
Possible data sources

The methods of automatically measuring semantic similarity fall into two large groups:

2. Extracting semantics from usage patterns in text corpora (distributional approach). Bottom-up.

We are interested in the second approach: semantics can be derived from the contexts a given word takes. Word meaning is typically defined by lexical co-occurrences in a large training corpus: count-based distributional semantics models.
Possible data sources

The methods of automatically measuring semantic similarity fall into two large groups:

- Building **ontologies** (knowledge-based approach). Top-down.
Intro: why semantic similarity?

Possible data sources

The methods of automatically measuring semantic similarity fall into two large groups:

- Building **ontologies** (knowledge-based approach). Top-down.
- Extracting semantics from **usage patterns** in text corpora (distributional approach). Bottom-up.

Word meaning is typically defined by lexical co-occurrences in a large training corpus: count-based distributional semantics models.
Possible data sources

The methods of automatically measuring semantic similarity fall into two large groups:

- Building **ontologies** (knowledge-based approach). Top-down.
- Extracting semantics from **usage patterns** in text corpora (distributional approach). Bottom-up.

We are interested in the second approach: semantics can be derived from the contexts a given word takes.
Possible data sources

The methods of automatically measuring semantic similarity fall into two large groups:

- Building ontologies (knowledge-based approach). Top-down.
- Extracting semantics from usage patterns in text corpora (distributional approach). Bottom-up.

We are interested in the second approach: semantics can be derived from the contexts a given word takes. Word meaning is typically defined by lexical co-occurrences in a large training corpus: count-based distributional semantics models.
Intro: why semantic similarity?

Similar words are close to each other in the space of their typical co-occurrences.
Contents

1 Intro: why semantic similarity?
2 Going neural
3 Task description
4 Texts in: used corpora
5 Meaning out: evaluation
6 Twiddling the knobs: importance of settings
7 Russian National Corpus wins
8 What next?
9 Q and A
Imitating the brain

Neurons receive signals with different weights from other neurons. Then they produce output depending on signals received. Artificial neural networks attempt to imitate this process.
Imitating the brain

- $10^{11}$ neurons in the brain, $10^4$ links connected to each.
Imitating the brain

- $10^{11}$ neurons in the brain, $10^4$ links connected to each.
- Neurons receive signals with different weights from other neurons.
Imitating the brain

- $10^{11}$ neurons in the brain, $10^4$ links connected to each.
- Neurons receive signals with different weights from other neurons.
- Then they produce output depending on signals received.
Imitating the brain

- $10^{11}$ neurons in the brain, $10^4$ links connected to each.
- Neurons receive signals with different weights from other neurons.
- Then they produce output depending on signals received.

Artificial neural networks attempt to imitate this process.
This paves way to so called *predict models* in distributional semantics.
This paves way to so called **predict models** in distributional semantics.

With **count models**, we calculate co-occurrences with other words and treat them as vectors.
This paves way to so called predict models in distributional semantics. With count models, we calculate co-occurrences with other words and treat them as vectors. With predict models, we use machine learning (especially neural networks) to directly learn vectors which maximize similarity between contextual neighbors found in the data, while minimizing similarity for unseen contexts.
This paves way to so called predict models in distributional semantics.

With count models, we calculate co-occurrences with other words and treat them as vectors.

With predict models, we use machine learning (especially neural networks) to directly learn vectors which maximize similarity between contextual neighbors found in the data, while minimizing similarity for unseen contexts.

The result is neural embeddings: dense vectors of smaller dimensionality (hundreds of components).
Going neural

Andrey Kutuzov  Igor Andreev (Mail.ru)

Texts in, meaning out: neural language models in semantic similarity task for Russian
Going neural
Going neural

Andrey Kutuzov, Igor Andreev (Mail.ru)

Texts in, meaning out: neural language models in semantic similarity tasks for Russian

28 May 2015, Dialogue, Moscow, Russia

философия

0.20
0.15
0.10
0.05
0.00
-0.05
-0.10
-0.15
-0.20
0
100
200
300
400
500

Andrey Kutuzov, Igor Andreev (Mail.ru)
There is evidence that concepts are stored in brain as neural activation patterns.
Going neural

There is evidence that concepts are stored in brain as neural activation patterns. Very similar to vector representations! Meaning is a set of distributed ‘semantic components’ which can be more or less activated.

NB: each component (neuron) is responsible for several concepts and each concept is represented by several neurons.
We employed Continuous Bag-of-words (CBOW) and Continuous Skip-gram (skip-gram) algorithms implemented in word2vec tool [Mikolov et al. 2013].
We employed Continuous Bag-of-words (CBOW) and Continuous Skip-gram (skip-gram) algorithms implemented in word2vec tool [Mikolov et al. 2013].

Shown to outperform traditional count DSMs in various semantic tasks for English [Baroni et al. 2014].
We employed Continuous Bag-of-words (CBOW) and Continuous Skip-gram (skip-gram) algorithms implemented in word2vec tool [Mikolov et al. 2013]. Shown to outperform traditional count DSMs in various semantic tasks for English [Baroni et al. 2014]. Is this the case for Russian?
We employed **Continuous Bag-of-words (CBOW)** and **Continuous Skip-gram** (skip-gram) algorithms implemented in *word2vec* tool [Mikolov et al. 2013]

- Shown to outperform traditional count DSMs in various semantic tasks for English [Baroni et al. 2014].
- Is this the case for Russian?
- Let’s train neural models on Russian language material.
RUSSE is the first attempt at semantic similarity evaluation contest for Russian language.
RUSSE is the first attempt at semantic similarity evaluation contest for Russian language. See [Panchenko et al 2015] and http://russe.nlpub.ru/ for details.
RUSSE is the first attempt at semantic similarity evaluation contest for Russian language. See [Panchenko et al 2015] and http://russe.nlpub.ru/ for details.

4 tracks:

1. hj (human judgment), relatedness task
2. rt (RuThes), relatedness task
3. ae (Russian Associative Thesaurus), association task
4. ae2 (Sociation database), association task
RUSSE is the first attempt at semantic similarity evaluation contest for Russian language. See [Panchenko et al 2015] and http://russe.nlpub.ru/ for details.

4 tracks:

1. hj (human judgment), relatedness task
2. rt (RuThes), relatedness task
3. ae (Russian Associative Thesaurus), association task
4. ae2 (Sociation database), association task

Participants presented with a list of word pairs; the task is to compute semantic similarity between each pair, in the range $[0;1]$. 
Task description

Comments on the shared task

1. rt and ae2 tracks: many related word pairs sharing long character strings (e.g., “благоразумие; благоразумность”). This allows reaching average precision of 0.79 for rt track and 0.72 for ae2 track using only character-level analysis.

2. Russian Associative Thesaurus (ae track) was collected between 1988 and 1997; lots of archaic entries (“колхоз; путь ильича”, “президент; ельцин”, etc). Not so for ae2 (Sociation.org database).
Task description

Comments on the shared task

1. **rt** and **ae2** tracks: many related word pairs sharing long character strings (e.g., “благоразумие; благоразумность”). This allows reaching average precision of 0.79 for **rt** track and 0.72 for **ae2** track using only character-level analysis.
Task description

Comments on the shared task

1. **rt** and **ae2** tracks: many related word pairs sharing long character strings (e.g., “благоразумие; благоразумность”). This allows reaching average precision of 0.79 for **rt** track and 0.72 for **ae2** track using only character-level analysis.

2. Russian Associative Thesaurus (**ae** track) was collected between 1988 and 1997; lots of archaic entries (“колхоз; путь ильича”, “президент; ельцин”, etc). Not so for **ae2** (Sociation.org database).
Contents

1 Intro: why semantic similarity?
2 Going neural
3 Task description
4 Texts in: used corpora
5 Meaning out: evaluation
6 Twiddling the knobs: importance of settings
7 Russian National Corpus wins
8 What next?
9 Q and A
### Texts in: used corpora

<table>
<thead>
<tr>
<th>Corpus</th>
<th>Size, tokens</th>
<th>Size, documents</th>
<th>Size, lemmas</th>
</tr>
</thead>
<tbody>
<tr>
<td>News</td>
<td>1300 mln</td>
<td>9 mln</td>
<td>19 mln</td>
</tr>
<tr>
<td>Web</td>
<td>620 mln</td>
<td>9 mln</td>
<td>5.5 mln</td>
</tr>
<tr>
<td>Ruscorpora</td>
<td>107 mln</td>
<td>$\approx$70 thousand</td>
<td>700 thousand</td>
</tr>
</tbody>
</table>

Lemmatized with MyStem 3.0, disambiguation turned on. Stop-words and single-word sentences removed.
<table>
<thead>
<tr>
<th>Corpus</th>
<th>Size, tokens</th>
<th>Size, documents</th>
<th>Size, lemmas</th>
</tr>
</thead>
<tbody>
<tr>
<td>News</td>
<td>1300 mln</td>
<td>9 mln</td>
<td>19 mln</td>
</tr>
<tr>
<td>Web</td>
<td>620 mln</td>
<td>9 mln</td>
<td>5.5 mln</td>
</tr>
<tr>
<td>Ruscorpora</td>
<td>107 mln</td>
<td>≈70 thousand</td>
<td>700 thousand</td>
</tr>
</tbody>
</table>

Lemmatized with MyStem 3.0, disambiguation turned on. Stop-words and single-word sentences removed.
Contents

1 Intro: why semantic similarity?
2 Going neural
3 Task description
4 Texts in: used corpora
5 Meaning out: evaluation
6 Twiddling the knobs: importance of settings
7 Russian National Corpus wins
8 What next?
9 Q and A
Possible causes of model failing

- The model outputs **incorrect similarity values**.
The model outputs incorrect similarity values.

Solution: retrain.
Possible causes of model failing

- The model outputs incorrect similarity values.
  - Solution: retrain.

- One or both words in the presented pair are unknown to the model.
  - Solutions:
Meaning out: evaluation

Possible causes of model failing

- The model outputs incorrect similarity values.
  - Solution: retrain.

- One or both words in the presented pair are unknown to the model.
  - Solutions:
    - Fall back to the longest commons string trick (increased average precision in rt track by 2...5%)
Possible causes of model failing

- The model outputs **incorrect similarity values**.
  - Solution: retrain.

- One or both words in the presented pair are **unknown to the model**.
  - Solutions:
    - Fall back to the **longest commons string** trick (increased average precision in rt track by 2…5%)
    - Fall back to another model trained on noisier and larger corpus (for example, RNC + Web).
<table>
<thead>
<tr>
<th></th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Intro: why semantic similarity?</td>
</tr>
<tr>
<td>2</td>
<td>Going neural</td>
</tr>
<tr>
<td>3</td>
<td>Task description</td>
</tr>
<tr>
<td>4</td>
<td>Texts in: used corpora</td>
</tr>
<tr>
<td>5</td>
<td>Meaning out: evaluation</td>
</tr>
<tr>
<td>6</td>
<td><strong>Twiddling the knobs: importance of settings</strong></td>
</tr>
<tr>
<td>7</td>
<td>Russian National Corpus wins</td>
</tr>
<tr>
<td>8</td>
<td>What next?</td>
</tr>
<tr>
<td>9</td>
<td>Q and A</td>
</tr>
</tbody>
</table>
Things are complicated

*1* CBOW or skip-gram algorithm. Needs further research; in our experience, CBOW is generally better for Russian (and faster).

*2* Vector size: how many distributed semantic features (dimensions) we use to describe a lemma.

*3* Window size: context width.

*4* Topical (associative) or functional (semantic proper) models.

*5* Frequency threshold: useful to get rid of long noisy lexical tail.

There is no silver bullet: set of optimal settings is unique for each particular task.

Increasing vector size generally improves performance, but not always.
Things are complicated

NNLM performance hugely depends not only on training corpus, but also on training settings:

1. CBOW or skip-gram algorithm. Needs further research; in our experience, CBOW is generally better for Russian (and faster).
2. Vector size: how many distributed semantic features (dimensions) we use to describe a lemma.
4. Frequency threshold: useful to get rid of long noisy lexical tail.

There is no silver bullet: set of optimal settings is unique for each particular task.

Increasing vector size generally improves performance, but not always.
Things are complicated

NNLM performance hugely depends not only on training corpus, but also on training settings:

1. **CBOW** or **skip-gram** algorithm. Needs further research; in our experience, CBOW is generally better for Russian (and faster).
NNLM performance hugely depends not only on training corpus, but also on training settings:

1. **CBOB** or **skip-gram** algorithm. Needs further research; in our experience, CBOB is generally better for Russian (and faster).

2. **Vector size**: how many distributed semantic features (dimensions) we use to describe a lemma.
Things are complicated

NNLM performance hugely depends not only on training corpus, but also on training settings:

1. **CBOV** or **skip-gram** algorithm. Needs further research; in our experience, CBOV is generally better for Russian (and faster).

2. **Vector size**: how many distributed semantic features (dimensions) we use to describe a lemma.

3. **Window size**: context width. **Topical** (associative) or **functional** (semantic proper) models.

There is no silver bullet: set of optimal settings is unique for each particular task. Increasing vector size generally improves performance, but not always.
Things are complicated

NNLM performance hugely depends not only on training corpus, but also on training settings:

1. **CBOW** or **skip-gram** algorithm. Needs further research; in our experience, CBOW is generally better for Russian (and faster).

2. **Vector size**: how many distributed semantic features (dimensions) we use to describe a lemma.

3. **Window size**: context width. **Topical** (associative) or **functional** (semantic proper) models.

4. **Frequency threshold**: useful to get rid of long noisy lexical tail.
**Things are complicated**

NNLM performance hugely depends not only on training corpus, but also on training settings:

1. **CBOW or skip-gram** algorithm. Needs further research; in our experience, CBOW is generally better for Russian (and faster).
2. **Vector size**: how many distributed semantic features (dimensions) we use to describe a lemma.
3. **Window size**: context width. **Topical** (associative) or **functional** (semantic proper) models.
4. **Frequency threshold**: useful to get rid of long noisy lexical tail.

There is no silver bullet: **set of optimal settings is unique for each particular task.**

Increasing vector size generally improves performance, but not always.
Twiddling the knobs: importance of settings

Our best-performing models submitted to RUSSE:
Our best-performing models submitted to RUSSE:

<table>
<thead>
<tr>
<th>Track</th>
<th>hj</th>
<th>rt</th>
<th>ae</th>
<th>ae2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rank</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>Training settings</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CBOW on Ruscorpora + CBOW on Web</td>
<td>0.7187</td>
<td>0.8839</td>
<td>0.8995</td>
<td>0.9662</td>
</tr>
</tbody>
</table>

The Russian National Corpus is better in semantic relatedness tasks; larger Web and News corpora provide good training data for association tasks.
Our best-performing models submitted to RUSSE:

<table>
<thead>
<tr>
<th>Track</th>
<th>hj</th>
<th>rt</th>
<th>ae</th>
<th>ae2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rank</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>Training settings</td>
<td>CBOW on Ruscorpora + CBOW on Web</td>
<td>CBOW on Ruscorpora + CBOW on Web</td>
<td>Skip-gram on News</td>
<td>CBOW on Web</td>
</tr>
<tr>
<td>Score</td>
<td>0.7187</td>
<td>0.8839</td>
<td>0.8995</td>
<td>0.9662</td>
</tr>
</tbody>
</table>

Russian National Corpus is better in semantic relatedness tasks; larger Web and News corpora provide good training data for association tasks.
Twiddling the knobs: importance of settings

Russian National Corpus models performance in rt track depending on vector size.
Twiddling the knobs: importance of settings

Web corpus models performance in rt track depending on vector size.
Twiddling the knobs: importance of settings

News corpus models performance in rt track depending on window size.
Twiddling the knobs: importance of settings

Russian National Corpus models performance in ae2 track depending on window size.
Twiddling the knobs: importance of settings

Russian National Corpus models performance in ae track depending on window size.
Russian Associative Thesaurus is more syntagmatic?
Russian Wikipedia models performance in ae track depending on vector size: considerable fluctuations.
Twiddling the knobs: importance of settings

See more on-line

http://ling.go.mail.ru/misc/dialogue_2015.html

- Performance plots for all settings combinations;
- Resulting models, trained with optimal settings.
1 Intro: why semantic similarity?
2 Going neural
3 Task description
4 Texts in: used corpora
5 Meaning out: evaluation
6 Twiddling the knobs: importance of settings
7 Russian National Corpus wins
8 What next?
9 Q and A
Models trained on RNC outperformed their competitors, often with vectors of lower dimensionality: especially in semantic relatedness tasks.
Models trained on RNC outperformed their competitors, often with vectors of lower dimensionality: especially in semantic relatedness tasks.

The corpus seems to be representative of the Russian language: balanced linguistic evidence for all major vocabulary tiers.
Models trained on RNC outperformed their competitors, often with vectors of lower dimensionality: especially in semantic relatedness tasks.

The corpus seems to be representative of the Russian language: balanced linguistic evidence for all major vocabulary tiers.

Little or no noise and junk fragments.
- Models trained on RNC outperformed their competitors, often with vectors of lower dimensionality: especially in semantic relatedness tasks.
- The corpus seems to be representative of the Russian language: balanced linguistic evidence for all major vocabulary tiers.
- Little or no noise and junk fragments.
Models trained on RNC outperformed their competitors, often with vectors of lower dimensionality: especially in semantic relatedness tasks.

The corpus seems to be representative of the Russian language: balanced linguistic evidence for all major vocabulary tiers.

Little or no noise and junk fragments.
Contents

1 Intro: why semantic similarity?
2 Going neural
3 Task description
4 Texts in: used corpora
5 Meaning out: evaluation
6 Twiddling the knobs: importance of settings
7 Russian National Corpus wins
8 What next?
9 Q and A
What next?

- Comprehensive study of semantic similarity errors typical to different models
Comprehensive study of semantic similarity errors typical to different models

Corpora comparison (including diachronic) via comparing NNLMs trained on these corpora, see [Kutuzov and Kuzmenko 2015];

Clustering vector representations of words to get coarse semantic classes (inter alia, useful in NER recognition, see [Sienšcnik 2015]);

Using neural embeddings in search engines industry: query expansion, semantic hashing of documents, etc.

Recommendation systems;

Sentiment analysis.
What next?

- Comprehensive study of semantic similarity **errors** typical to different models
- **Corpora comparison** (including diachronic) via comparing NNLMs trained on these corpora, see [Kutuzov and Kuzmenko 2015];
- **Clustering** vector representations of words to get coarse semantic classes (*inter alia*, useful in NER recognition, see [Sienčnik 2015]);

Using neural embeddings in search engines industry: query expansion, semantic hashing of documents, etc

Recommendation systems;

Sentiment analysis.
What next?

- Comprehensive study of semantic similarity \textit{errors} typical to different models
- \textbf{Corpora comparison} (including diachronic) via comparing NNLMs trained on these corpora, see [Kutuzov and Kuzmenko 2015];
- \textbf{Clustering} vector representations of words to get coarse semantic classes (\textit{inter alia}, useful in NER recognition, see [Sienčnik 2015]);
- Using neural embeddings in search engines industry: query expansion, semantic hashing of documents, etc
What next?

- Comprehensive study of semantic similarity errors typical to different models
- Corpora comparison (including diachronic) via comparing NNLMs trained on these corpora, see [Kutuzov and Kuzmenko 2015];
- Clustering vector representations of words to get coarse semantic classes (*inter alia*, useful in NER recognition, see [Sienčnik 2015]);
- Using neural embeddings in search engines industry: query expansion, semantic hashing of documents, etc
- Recommendation systems;
What next?

- Comprehensive study of semantic similarity errors typical to different models
- **Corpora comparison** (including diachronic) via comparing NNLMs trained on these corpora, see [Kutuzov and Kuzmenko 2015];
- **Clustering** vector representations of words to get coarse semantic classes (*inter alia*, useful in NER recognition, see [Sienčnik 2015]);
- Using neural embeddings in search engines industry: query expansion, semantic hashing of documents, etc
- Recommendation systems;
- Sentiment analysis.
What next?

Distributional semantic models for Russian on-line

As a sign of reverence to RusCorpora project, we launched a beta version of Rus Vectores web service:

http://ling.go.mail.ru/dsm
Distributional semantic models for Russian on-line

As a sign of reverence to *RusCorpora* project, we launched a beta version of *RusVectōrēs* web service:

http://ling.go.mail.ru/dsm
What next?

http://ling.go.mail.ru/dsm:

- Find nearest semantic neighbors of Russian words;
What next?

http://ling.go.mail.ru/dsm:

- Find nearest semantic neighbors of Russian words;
- Compute cosine similarity between pairs of words;
http://ling.go.mail.ru/dsm:

- Find nearest semantic neighbors of Russian words;
- Compute cosine similarity between pairs of words;
- Perform algebraic operations on word vectors (‘крыло’ - ‘самолет’ + ‘машина’ = ‘колесо’);
http://ling.go.mail.ru/dsm:

- Find nearest semantic neighbors of Russian words;
- Compute cosine similarity between pairs of words;
- Perform algebraic operations on word vectors (‘крыло’ - ‘самолет’ + ‘машине’ = ‘колесо’);
- Optionally limit results to particular parts-of-speech;
What next?

http://ling.go.mail.ru/dsm:

- Find nearest semantic neighbors of Russian words;
- Compute cosine similarity between pairs of words;
- Perform algebraic operations on word vectors (‘крыло’ - ‘самолет’ + ‘машина’ = ‘колесо’);
- Optionally limit results to particular parts-of-speech;
- Choose one of five models trained on different corpora;
http://ling.go.mail.ru/dsm:

- Find nearest semantic neighbors of Russian words;
- Compute cosine similarity between pairs of words;
- Perform algebraic operations on word vectors (‘крыло’ - ‘самолет’ + ‘машина’ = ‘колесо’);
- Optionally limit results to particular parts-of-speech;
- Choose one of five models trained on different corpora;
- ... or upload your own corpus and have a model trained on it;
What next?

http://ling.go.mail.ru/dsm:

- Find nearest semantic neighbors of Russian words;
- Compute cosine similarity between pairs of words;
- Perform algebraic operations on word vectors (‘крыло’ - ‘самолет’ + ‘машина’ = ‘колесо’);
- Optionally limit results to particular parts-of-speech;
- Choose one of five models trained on different corpora;
- ... or upload your own corpus and have a model trained on it;
- Every lemma in every model is identified by a unique URI:
What next?

http://ling.go.mail.ru/dsm:

- Find nearest semantic neighbors of Russian words;
- Compute cosine similarity between pairs of words;
- Perform algebraic operations on word vectors (‘крыло’ - ‘самолет’ + ‘машина’ = ‘колесо’);
- Optionally limit results to particular parts-of-speech;
- Choose one of five models trained on different corpora;
- ... or upload your own corpus and have a model trained on it;
- Every lemma in every model is identified by a unique URI:
  - http://ling.go.mail.ru/dsm/ruscorpora/диалог ;
What next?

http://ling.go.mail.ru/dsm:

- Find nearest semantic neighbors of Russian words;
- Compute cosine similarity between pairs of words;
- Perform algebraic operations on word vectors (‘крыло’ - ‘самолет’ + ‘машина’ = ‘колесо’);
- Optionally limit results to particular parts-of-speech;
- Choose one of five models trained on different corpora;
- ... or upload your own corpus and have a model trained on it;
- Every lemma in every model is identified by a unique URI:
  - http://ling.go.mail.ru/dsm/ruscorpora/диалог;
- Creative Commons Attribution license;
- More to come!
Contents

1 Intro: why semantic similarity?
2 Going neural
3 Task description
4 Texts in: used corpora
5 Meaning out: evaluation
6 Twiddling the knobs: importance of settings
7 Russian National Corpus wins
8 What next?
9 Q and A
Thank you!

Questions are welcome.

Texts in, meaning out: neural language models in semantic similarity task for Russian

Andrey Kutuzov (akutuzov@hse.ru)
Igor Andreev (i.andreev@corp.mail.ru)

28 May 2015
Dialogue, Moscow, Russia