© 2018 г. Р.В. Кузнецова (kuznetsova@ap-team.ru)
О.Ю. Бахтеев (bakhteev@ap-team.ru)
(Компания Антиплагиат, Московский физико-технический институт),
Ю.В. Чехович, канд. физ.-мат.наук (chehovich@ap-team.ru)
(Вычислительный центр им. А.А. Дородницына ФИЦ ИУ РАН)

Детектирование переводных заимствований в больших массивах научных документов[1]

Рассматривается задача детектирования переводных заимствований для пары языков русский-английский. Для решения предлагается использовать моноязыковой подход — сводить задачу детектирования заимствований к одному языку, используя машинный перевод. В связи со спецификой рассматриваемой задачи, предлагаемый алгоритм детектирования должен быть устойчив к неоднозначностям перевода. Сначала отбираются документы-кандидаты, и эта устойчивость достигается за счет замены слов на метки классов эквивалентности, полученные с помощью дистрибутивной модели. Затем происходит сравнения найденных кандидатов и рассматриваемого документа, для этого используется отображение текстовых фрагментов документов в векторное пространство высокой размерности. Вычислительный эксперимент проводится на двух выборках — сгенерированном корпусе и на статьях из журналов, входящих в Российский индекс научного цитирования (РИНЦ).

Ключевые слова: автоматическая обработка текстов, машинный перевод, глубокое обучение, переводные заимствования, обнаружение переводных заимствований, дистрибутивная семантика.

1. Введение

Проблема некорректных текстовых заимствований актуальна для сферы образования и научных исследований [1]. По информации ряда вузов, не менее 50 процентов дипломных работ, защищенных на «отлично», в 2005 году были скопированы из интернета без изменений. По материалам исследования, проведенного в 2013 году компанией Антиплагиат по заказу РГБ[2], более 1500 диссертаций по историческим наукам, защищенных в России после 2000 года, содержат значительные заимствования из других диссертаций [5].

Для задачи обнаружения заимствований в рамках одного языка высокую полноту поиска показывают промышленные инструменты [1], работа которых основана на алгоритме построения инвертированного индекса [8, 9], где документ из коллекции представляется в виде набора перекрывающих друг друга пословных n -грамм

[1]Работа выполнена при поддержке Российского фонда фундаментальных исследований (проект 18-07-01441).

[2]http://olden.rsl.ru/en

(шинглов). Для поиска кандидатов для заданного документа проводится поиск по инвертированному индексу с дальнейшим упорядочиванием кандидатов в соответствие с выбранной функцией схожести совпавших шинглов. В работах [10, 11] описываются алгоритмы поиска, основанные на извлечении ключевых слов. Преимущество таких алгоритмов состоит в простоте реализации и в вычислительной эффективности. Однако, у существующих промышленных систем существует уязвимость — они не могут обнаруживать заимствования из другого языка.

Существует несколько подходов, описывающих эту проблему для некоторых пар языков [2, 3, 4], например для пары испанский-английский. В статье [2] замечено, что качество алгоритмов детектирования переводных заимствований сильно зависит от рассматриваемой пары языков и степени их родства. Поэтому задача остается по прежнему актуальной — в данной работе рассматривается пара русский-английский, которая не входит в одну лингвистическую группу. Это создает дополнительные трудности при разработке алгоритма детектирования.

Выбор пары языков русский-английский обусловлен преобладанием англоязычных публикаций в интернете и лучшим знанием этого языка по сравнению с другими. Количество публикаций на английском, проиндексированных такими базами данных как Web of Science и Scopus, подтверждают этот факт. Еще одним важным фактором является развитие систем машинного перевода, использование которых позволит получить «оригинальную» работу, не прикладывая практически никаких усилий. И, как упоминалось раннее, заимствовать в рамках одного языка становится все сложнее, значит нужен другой способ обхода промышленных систем.

Алгоритм детектирования переводных заимствований для пары языков русский-английский практически не встречается в литературе из-за сложной грамматики русского языка и нехватки данных конкретно для этого случая.

Как и в работах [6, 7] в данной статье предлагается описание всего алгоритма — сначала ведется поиск документов-кандидатов по внешней коллекции, затем происходит их детальное сравнение с проверяемым документом.

Важным требованием к разрабатываемому алгоритму является устойчивость к неоднозначностям перевода — машинного или ручного. Алгоритмы, используемые в промышленных решениях, являются неустойчивыми к перестановке слов и обычно используются для детектирования почти -дубликатов. Поэтому, для этапа поиска текстов-кандидатов, в данной статье используется модификация алгоритма шинглов: каждому слову в шингле ставится в соответствие класс эквивалентности, полученный на основе кластеризации векторов слов с использованием моделей дистрибутивной гипотезы.

Многие работы по теме обнаружения переводных заимствований рассматривают для сравнения документов между собой решения, основанные на двуязычных или моноязычных векторах слов [12, 13], но практически никто не использует векторы фраз. Предлагается алгоритм, основанный на моноязыковом анализе документов — проверяемый документ переводится на английский язык с использованием системы машинного перевода. Так как решение должно быть устойчивым к неоднозначностям перевода, предлагается сравнивать между собой не сами текстовые фрагменты, а векторы, им соответствующие. Так как мы используем моноязыковой подход, предлагается рассматривать перевод как частный случай перефразировки текста [14]. В данном предположении становится возможным оптимизация модели

векторного представления фраз с использованием существующих англоязычных выборок по определению перефразировок.

Рассматривается случай, когда проверяемый документ написан на русском языке и содержит вставки текста, переведенного с английского языка. Этот случай особенно важен для научных работ, так как введение, обзор литературы и даже основная часть часто содержат текстовые заимствования. Общая схема алгоритма приведена на рисунке (1) и включает следующие шаги:

1) *Машинный перевод* — перевод проверяемого документа на английский язык. Для этого используется система статистического машинного перевода Moses [15] с открытым исходным кодом.

2) *Поиск документов-кандидатов* — для проверяемого документа находятся наиболее релевантные документы-кандидаты, для этого используется модификация алгоритма шинглов. Детали представлены в (4).

3) *Модель векторного представления фразы* —текст разбивается на фразы и строится отображение каждой фразы в векторное пространство. Для построения этого отображения используются автокодировщики, которые показывает высокое качество решения этой задачи [16, 17]. Для оптимизации параметров автокодировщика применяются алгоритмы обучения без учителя и алгоритмы частичного обучения. Вводится составная функция ошибки. Детали представлены в (5).

4) *Сравнение документов* — используется алгоритм приближенного поиска ближайшего соседа для того, чтобы найти ограниченное количество близких фраз. Детали представлены в (5.1).

5) *Пост-обработка текста* — после сравнения текстовых фрагментов используется алгоритм классификации для снижения доли ложно-положительных срабатываний. Детали представлены в (5.1).

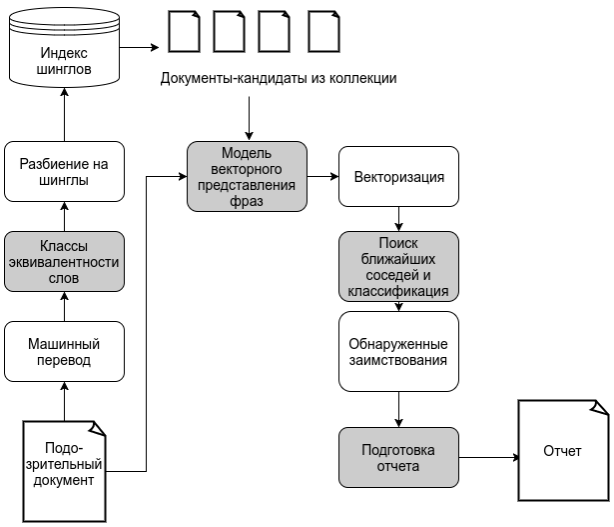


Рис. 1: Схема алгоритма.

Для вычислительного эксперимента предлагается методика генерации выборки для пары языков русский-английский. Методика описана в разделе (6.1). Вычислительный эксперимент проводится на двух выборках — сгенерированном корпусе (6.3) и на статьях из журналов, входящих в Российский индекс научного цитирования (РИНЦ). Результаты эксперимента и анализ ошибок представлены в разделе (7).

2. Обзор литературы

Предлагаемый алгоритм близок к рассматриваемым в работах [18, 19] подходами, где проверяемый документ переводится на язык внешней коллекции с помощью систем машинного перевода. В работе [20] модель перевода IBM-1 используется для извлечения информации о близости между текстами. Авторы работ [21, 22] предлагают алгоритмы, основанные на использовании n -грамм и на статистике используемых терминов.

В ряде работ используются дополнительные ресурсы, такие как тезаурусы и онтологии. В работах [12, 3, 4] авторы предлагают использовать BabelNet [23] и WordNet [24] для извлечения информации о близости между текстами. В работе [12] предлагается алгоритм, основанный на комбинации нейронных сетей и графов знаний. В некоторых работах [25, 26, 27] для определения близости между текстами на разных языках используются алгоритмы близкие к латентному семантическому анализу [28], использующие разложение матрицы слово-документ. Текущий общепризнанный подход [29, 12], показывающий высокое качество основан на использовании семантических графов для каждого документа. Близость между текстами оценивается как близость между соответствующими графами. Основным недостатком этого подхода является ресурсоемкость: использование мультиязычных онтологий, таких как BabelNet [23], требует больших вычислительных мощностей для построения семантических графов для каждого текстового фрагмента, а также сравнения полученных семантических гарфов.

Другой класс работ посвящен поиску документов-кандидатов. В работах [10, 9] производится сравнение нескольких алгоритмов поиска. Ряд работ для этой задачи [30, 31] предлагает использовать векторы параграфов или документов. Одной из проблем таких алгоритмов является вычислительная дороговизна. В работах [32] предлагается алгоритм приближенного поиска ближайшего соседа, который позволяет осуществлять поиск документов быстрее за счет большей нагрузки на память.

Так как в предлагаемом алгоритме используется моноязыковой анализ, задача близка к задаче детектирование перефразированного текста. Многие подходы [33, 34, 17] для решения этой задачи используют векторы фраз, полученные с помощью нейронных сетей глубокого обучения. В работе [35] предлагается Neural Bag-of-Words и deep averaging networks. В работах [33, 36] авторы предлагают рекурсивные нейронные сети, построенные по структуре деревьев разбора, полученных с помощью грамматик зависимостей и грамматик составляющих. В работах [17, 37] авторы предлагают использовать LSTM [38] и GRU [39]. В работах [40] предлагается иерархический LSTM для построения модели векторных представлений параграфов текста. Для задачи детектирования перефразированного текста в случае разных языков также используются алгоритмы, основанные на использовании двуязычных автокодировщиков [41, 42] или сиамских нейронных сетей [34]. В противовес работам [17, 43, 44] в данной статье предлагается использовать выходы нейронной сети

как векторные представления фраз для дальнейшего приближенного алгоритма поиска ближайшего соседа [45].

Работа посвящена обнаружению переводных заимствований для пары языков русский-английский. Данная пара не является часто встречаемой в литературе, насколько известно, существует только одна работа, посвященная решению данной задачи [19]. В данной работе также предлагается методика генерации корпуса, содержащего переводные заимствования. Данная работа может послужить основой для дальнейших исследований способов обнаружения переводных заимствований для этой языковой пары.

3. Формальная постановка задачи

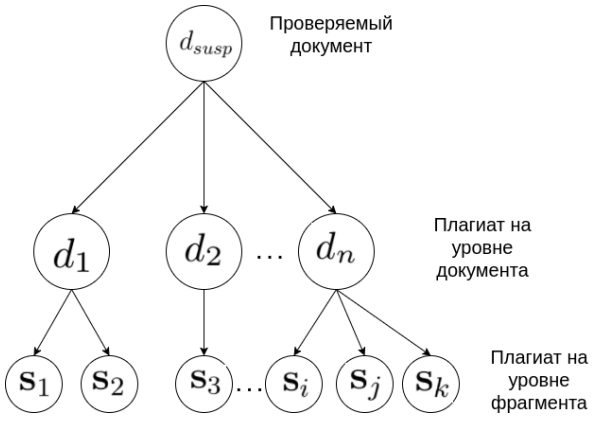
Введем некоторые обозначения:

- Коллекция документов D — неупорядоченное множество документов.
- Документ d_k — упорядоченное множество фрагментов s_i.
- Фрагмент s_i — упорядоченное множество слов $\{x_1, x_2, \ldots, x_n\}$.

Введем иерархическую структуру процесса обнаружения заимствований. На первом уровне иерархии будем определять заимствования на *уровне документа*, т.е находить те документы $d_i \in D$ коллекции, из которых есть некоторое количество заимствованных фрагментов s_i в проверяемом документе d_{susp}, но на данном уровне иерархии эти фрагменты не идентифицируются. Будем называть эти документы документами-кандидатами.

На втором уровне иерархии определяются заимствования на *уровне фрагмента*,т.е в каждом найденном документе d_i требуется идентифицировать конкретные заимствованные фрагменты s_i.

Обе задачи ставятся как задачи информационного поиска, их формальные постановки приведены в разделах (3.1) и (3.2).



3.1. Поиск документов кандидатов

Решается задача обнаружения заимствований на первом уровне иерархии — к каждому документу d^i_{susp} требуется найти те документы $d^j \in D$, из которых были произведены заимствования.

Задана коллекция документов на английском языке $D = \{d^j\}$ и коллекция документов, являющихся переводом с русского на английский язык $D_{susp} = \{d^i_{susp}\}$.

Коллекции D задается множество запросов — проверяемых переведенных документов $D_{susp} = \{d^i_{susp}\}$. Для каждого документа-запроса $d^i_{susp} \in D_{susp}$ документы подмножества коллекции $D_q \subset D$ экспертно оценены, т.е существует бинарное отношение информационного поиска:

$$g_{doc} : D_{susp} \times D_q \to \{0, 1\}, \tag{1}$$

где 1 соответствует релевантному найденному документу, а 0- нерелевантному. Цель — найти функцию f_{doc}, аппроксимирующую отношение g_{doc}.

Аргумент искомой функции f — признак пары (документ-запрос, документ коллекции):

$$\varphi^\alpha(d, d_{susp}) = \sum_{h \in \mathcal{H}(d_{susp})} \frac{\mathbf{I}[h \in \mathcal{H}(d)]}{|d' \in C : h \in \mathcal{H}(d')|^\alpha},$$

где $\mathcal{H}$ — функция перевода документа во множество n -грамм, упорядоченную последовательность n слов, $\alpha \in \mathbb{R}$.

Качество модели оценивается с помощью функционала качества Recall:

$$\text{Recall} = \frac{1}{|D_{susp}|} \sum_{d_{susp} \in D_{susp}} \frac{|\text{relevant}(d_{susp}) \cap \text{retrieved}(d_{susp})|}{|\text{relevant}(d_{susp})|},$$

где

$$\text{relevant}(d_{susp}) = \{d \in D_q : g_{doc}(d_{susp}, d) = 1\},$$
$$\text{retrieved}(d_{susp}) = \{d \in D_q : f_{doc}(d_{susp}, d) = 1\}.$$

Функция f — решение задачи максимизации:

$$\hat{f}_{doc} = \arg \max_{f_{doc} \in \mathcal{F}} (\text{Recall}(f_{doc}, g_{doc}, D_{susp}, D)). \tag{2}$$

Решение выбирается из множества $\mathcal{F}$ функций вида $f_{doc}(\varphi^\alpha)$.

Предлагаемый алгоритм для решения задачи (2) приведен в разделе (4).

3.2. Сравнение документов

На втором уровне иерархии для каждого проверяемого документа d_{susp} и множества найденных к нему на первом уровне иерархии $\text{retrieved}(d_{susp})$.

Пусть $S_{susp} = \{s^l_{susp}\}$ и $S_{retr} = \{s^k_{retr}\}$ — множество текстовых фрагментов, полученных в результате разбиения d_{susp} и $\text{retrieved}(d_{susp}))$ соответственно. Как и в задаче (1), экспертно задано бинарное отношение между фрагментами множества S_{susp} и подмножеством $S_{retr_q} \subset S_{retr}$:

$$g_{frag} : S_{susp} \times S_{retr_q} \to \{0, 1\},$$

где 1 соответствует релевантному найденному фрагменту, а 0- нерелевантному.
Определим множество релевантных фрагментов для фрагмента проверяемого документа $s_{susp} \in S_{susp}$:

$$\text{relevant}(s_{susp}) = \{s_{retr} \in S_{retr_q} : g_{frag}(s_{susp}, s_{retr}) = 1\}.$$

Определим множество найденных фрагментов для фрагмента проверяемого документа $s_{susp} \in S_{susp}$:

$$\text{retrieved}(s_{susp}) = \{s_{retr} \in S_{retr_q} : f_{frag}(s_{susp}, s_{retr}) = 1\},$$

где функция f_{frag} аппроксимирует отношение g_{frag}.

Требуется найти функцию $f_{frag} \in \mathcal{G}$:

$$\hat{f}_{frag} = \arg\max_{f_{frag} \in \mathcal{G}}(F1(f_{frag}, g_{frag}, S_{susp}, S_{retr})). \tag{3}$$

где $\mathcal{G}$ — множество моделей рекуррентных автокодировщиков.
F1 -мера:

$$F1 = \frac{2PR}{P+R},$$

P — precision, R — recall:

$$P = \frac{|\text{relevant}(s_{susp}) \cap \text{retrieved}(s_{susp})|}{|\text{retrieved}(s_{susp})|}, R = \frac{|\text{relevant}(s_{susp}) \cap \text{retrieved}(s_{susp})|}{|\text{relevant}(s_{susp})|}.$$

Предлагаемый алгоритм для решения задачи (3) приведен в разделе (5).

4. Поиск документов-кандидатов

Основная проблема при поиске документов-кандидатов заключается в том, что этап машинного перевода порождает тексты, которые могут существенно отличаться от источника заимствования. Ниже приводится пример таких текстов.

- Having considered the dimensions next the policy analyst has to identify various indicators for each dimension.
- Having considered the size of the following political analyst should identify the different indicators for each measurement.

Одним из алгоритмов поиска документов-кандидатов в задачах обнаружения дословных заимствований и поиска почти-дубликатов текста является построения инвертированного индекса, в котором каждый документ коллекции представляется набором *шинглов* [9], т.е. набором перекрывающихся n-грамм. Проверяемый документ также разбивается на шинглы, после чего проводится поиск документов по инвертированному индексу с наибольшим совпадением шинглов. В то же время, в случае поиска документов-кандидатов в задаче детектирования переводных заимствований данный алгоритм может показывать слабое качество в силу неоднозначности перевода.

Для уменьшения влияния неоднозначности перевода на поиск документов-кандидатов предлагается проводить предварительное разбиение слов на классы эквивалентности и замену слов на соответствующие им метки классов: $\{x_1, \ldots, x_n\} \to \{\text{class}(x_1), \ldots, \text{class}(x_n)\}$,
где $x_1, \ldots, x_n$ — слова, class — функция перевода слов в классы эквивалентности.

Каждый класс эквивалентности содержит семантически близкие слова. Также для уменьшения неоднозначности перевода перед разбиением на n-граммы предлагается удалять из текста стоп-слова и проводить лемматизацию. Для учета возможных перестановок слов, возникающих после перевода текста, слова внутри каждой n-граммы сортируются в лексикографическом порядке.

В данной работе для получения классов эквивалентности используется модель векторного представления слов, работа которой основана на дистрибутивной гипотезе. Формирование классов эквивалентности заключается в кластеризации векторов, полученных с использованием данной модели, с использованием косинусной меры как кластерной меры близости.

Ниже приведены примеры полученных классов эквивалентности:

- *[beer, beers, brewing, ale, brew, brewery, pint, stout, guinness, ipa, brewed, lager, ales, brews, pints, cask]*
- *[survey, assessment, evaluation, evaluate, examine, assess, surveys, analyze, evaluating, assessments, examining, analyzing, assessing, questionnaire, evaluations, analyse, questionnaires, analysing]*
- *[brilliant, excellent, exceptional, finest, outstanding, super, terrific]*

Результаты эксперимента по поиску документов-кандидатов представлены в разделе (6.3).

5. Сравнение документов

Для сравнения между найденными документами-кандидатами и проверяемым документом используется модель векторного представления фразы — тексты разбиваются на фрагменты и сравниваются соответствующие им векторы. Пусть задана выборка $S = \{s_i\}$, состоящая из фраз $s_i = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ где $\mathbf{x}_k \in \mathbf{X}$ вектор слова. Требуется найти отображение:

$$\phi : s_i \to \mathbf{s}_i \in \mathbb{R}^n,$$

такое, что

$$Q(S, \phi) = \arg\max_{\phi_\Phi} |\cos(\phi(s_i), \phi(s_j)) - \cos(\phi(s_i), \phi(s_k))|,$$

$(s_i, s_j) \in \mathcal{S}$ — выборка пар схожих фрагментов, $(s_i, s_k) \in \mathcal{D}$ — выборка пар несхожих фрагментов, Φ — множество рекуррентных моделей автокодировщиков. Следует отметить, что подмодульное выражение всегда неотрицательно, в силу способа построения выборок $\mathcal{S}$ и $\mathcal{D}$.

Первое слагаемое оптимизируемой функции ошибки — ошибка реконструкции между входным и выходным вектором. Используется GRU-GRU автокодировщик [39].
GRU —кодирующий блок:

$$\begin{aligned} \mathbf{z}_t &= \sigma(\mathbf{W}_z \mathbf{x}_t + \mathbf{W}_z \mathbf{h}_{t-1} + \mathbf{b}_z), \\ \mathbf{r}_t &= \sigma(\mathbf{W}_r \mathbf{x}_t + \mathbf{W}_r \mathbf{h}_{t-1} + \mathbf{b}_r), \\ \tilde{\mathbf{h}}_t &= \tanh(\mathbf{W}_x \mathbf{x}_t + \mathbf{W}_h \mathbf{h}_{t-1} + \mathbf{b}), \\ \mathbf{h}_t &= (1 - \mathbf{z}_t) \circ \mathbf{h}_{t-1} + \mathbf{z}_t \circ \tilde{\mathbf{h}}_t. \end{aligned} \tag{4}$$

Выход кодирующего блока — вектор $\mathbf{h}_e^n$, соответствующий всей входной фразе, используется как начальное состояние декодирующего блока, т.е. $\mathbf{h}_e^n = \mathbf{h}_e^d$. Используя линейное преобразование, предсказывается следующий вектор слова $\hat{\mathbf{x}}_n = \mathbf{W}_d\mathbf{h}_t + \mathbf{b}_d$. На дальнейших шагах декодирующий блок использует предсказанный вектор слова $\hat{\mathbf{x}}_n$ и скрытое состояние $\mathbf{h}_{t-1}$. Минимизируется ошибка реконструкции:

$$E_{rec} = \sum_{i=1}^{n} \parallel \mathbf{x}_i - \hat{\mathbf{x}}_i \parallel_2^2 . \tag{5}$$

Финальное скрытое состояние $\mathbf{h}_e^n$ соответствует всей фразе $\mathbf{s}_i$.
Второе слагаемое составной функции ошибки — ошибка отступа. Для оптимизации этой функции ошибки используется выборка $\mathcal{S} = \{(s_i, s_j)\}$, состоящая из пар схожих фраз как в [37].

$$E_{me} = \frac{1}{|\mathcal{S}|}\left(\sum_{(s_i,s_j)\in\mathcal{S}} \max(0, \delta - c_-) + \max(0, \delta - c_+)\right), \tag{6}$$

where $c_- = \cos(\mathbf{s}_i, \mathbf{s}_j) - \cos(\mathbf{s}_i, \mathbf{s}_{i'})$, $c_+ = \cos(\mathbf{s}_i, \mathbf{s}_j) + \cos(\mathbf{s}_j, \mathbf{s}_{j'})$, δ — отступ,
$\mathbf{s}_{i'} = \arg\max_{\mathbf{s}_{i'}\in\mathcal{S}_b\setminus(\mathbf{s}_i,\mathbf{s}_j)} \cos(\mathbf{s}_i, \mathbf{s}_{i'})$, $\mathcal{S}_b \in \mathcal{S}$ — текущий батч.
Итоговая функция ошибки:

$$\alpha E_{rec} + (1 - \alpha)E_{me} \to \min, \tag{7}$$

где α — настраиваемый гиперпараметр.

5.1. Классификатор

Для каждого вектора фразы $\mathbf{s}_i$ из проверяемого документа d_{susp}^i находится r ближайших векторов по косинусной мере из документов-кандидатов retrieved(d_{susp}) используя библиотеку *Annoy*[3]. Основная цель этого — сократить количество пар фрагментов для классификации.

Для пары (s_i, s_j) и соответствующих этой паре векторов $(\mathbf{s}_i, \mathbf{s}_j)$ рассматривается следующее решающее правило:

$$f(s_i, s_j) = \begin{cases} 1, \text{ если } \cos(\mathbf{s}_i, \mathbf{s}_j) > t_1 \text{ and } p(\mathbf{s}_i, \mathbf{s}_j) > t_2, \\ 0 \text{ иначе,} \end{cases}$$

где p — вероятность классификатора, t_1 — порог косинусной меры и t_2 — минимальный порог вероятности классификатора[4]. Гистограмма косинусных мер между парами фрагментов и величиной t_1 приведена на рис. 2.

В качестве признаков используется конкатенация разницы по модулю и покомпонентное произведение компонент вектора $[|\mathbf{s}_i - \mathbf{s}_j|, \mathbf{s}_i * \mathbf{s}_j]$. В качестве классификатора выступает Random Forest.

[3] https://github.com/spotify/annoy
[4] $t_1 = 0.6, t_2 = 0.5$.

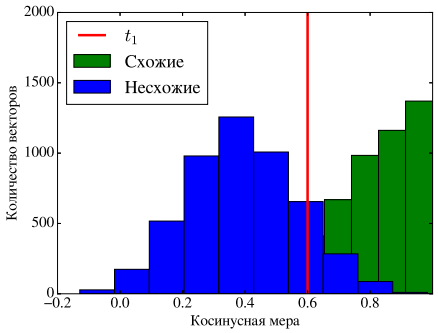


Рис. 2: Гистограмма косинусных мер между парами схожих и несхожих фраз

6. Вычислительный эксперимент

Для анализа качества предложенного алгоритма был проведен ряд вычислительных экспериментов как на сгенерированных выборках, так и на реальных коллекциях документов. В данном разделе приводятся детали порождения сгенерированных выборок, и эксперименты, проведенные на них.

6.1. Сгенерированная коллекция переводных заимствований

Для порождения переводных заимствований были использованы документы из английской и русской версии сайта Wikipedia.

В качестве коллекции документов D были использованы 100 тыс. статей из английской версии Wikipedia. В качестве коллекции проверяемых документов D_{susp} использовалась случайная подвыборка документов из русской версии Wikipedia. Для порождения заимствований для каждого документа $d^i_{susp} \in D_{\text{susp}}$ использовался следующий алгоритм:

1) Выбрать документы-кандидаты $\{d^i\}$ из коллекции D. Для уменьшения разброса лексики в документах-кандидатах и проверяемом документе выбор документов-кандидатов производился из подвыборки 500 наиболее релевантных документов для проверяемого документа d^i_{susp}. Для определения релевантности использовалась *tf· idf*-мера. Количество документов-кандидатов выбиралось случайно от 1 до 10.

2) Выбрать предложения из документов-кандидатов $\{d^i\}$ случайным образом и перевести их на русский язык.

3) Заменить случайные предложения из проверяемого документа d^i_{susp} на переведенные предложения из документов-кандидатов. Процент замененных предложений из проверяемого документа d^i_{susp} выбирался случайно от 20 до 80%.

Полученная выборка доступна по адресу[5].

[5] http://tiny.cc/cl_ru_en

6.2. Оптимизация параметров рассматриваемых моделей

В качестве модели векторного представления слов использовалс библиотека `fastText`, оптимизация параметров которой проводилась на английской версии Wikipedia. Размерность векторного пространства для векторного представления слов и фраз была установлена как 100[6]. Для построения классов эквивалентности была использована агломеративная кластеризация на векторах слов. В качестве меры близости слов рассматривалась косинусная мера между соответствующими векторными представлениями. Итоговая модель классов эквивалентности содержала 30 тыс. классов доля 777 тысяч слов.

В качестве системы машинного перевода использовался Moses [15], модель которого была была обучена на 18.5 миллионах параллельных предложений из корпусов Opus [46]. В качестве языковой модели для системы машинного перевода использовалась 3-граммная модель 3-gram IRSTLM [47].

В качестве выборки для минимизации ошибки реконструкции E_{rec} (5) использовались 10 миллионов предложений из английской версии Wikipedia. Второе слагаемое функции потерь (7) использует инфомрацию о похожих предложениях $\mathcal{S} = \{(s_i, s_j)\}$. В качестве выборки таких предложений использовались пары параллельных предложений из корпуса OpenSubtitles [46]. Ниже приведены примеры полученных предложений:

- *You know, I remember you pitched me the idea for this thing five years ago.*
- *I remember you pitched me the idea for this to the cause of 5 years ago.*

Для оптимизации параметров классификатора также использовалась подвыборка данного корпуса. Выбирались пары похожих предложений с высокой косинусной мерой и пары непохожих предложений с низкой косинусной мерой.

6.3. Детали вычислительного эксперимента

Было проведено три эксперимента на сгенерированных данных:

1) Поиск кандидатов. В данном эксперименте анализировалось качество полученной модели классов эквивалентности слов. В качестве базового эксперимента для сравнения рассматривался алгоритм, основанный на шинглах без приведения слов к классам эквивалентности.

2) Сравнение фрагментов текста. В данном эксперименте рассматривался случай, когда отбор кандидатов был проведен полностью корректно: Recall@10 = 1.0. В качестве базового алгоритма также выступал алгоритм, основанный на шинглах: проверяемый документ d^i_{susp} переводился на английский язык. После этого полученный текст проходил лемматизацию и разбивался на множество перекрывающихся 4-грамм. Для учета возможных перестановок слов при переводе, слова внутри каждой 4-граммы сортировались. Результатом сравнения двух документов выступало множество совпавших отсортированных 4-грамм.

[6]Для оптимизации модели векторного представления фраз использовался алгоритм AdaDelta с параметрами $\epsilon = 10^{-6}, \mu = 0.95$ и L2-регуляризация $\lambda_2 = 10^{-6}$. Для итоговой фукнции потерь (7) были установлены следующие значения гиперпараметров: $\delta = 0.3, \alpha = 0.1$.

3) Эксперимент, оценивающий качество всего алгоритма (поиск кандидатов и сравнение фрагментов текста). Данный эксперимент позволял оценить качество представленного алгоритма в целом.

Результаты эксперимента по поиску кандидатов представлены в табл. 1. Представленный алгоритм, основанный на построении моделей классов эквивалентности, дает лучшее качество, чем базовый алгоритм, основанный на шинглах.

Алгоритм	**Recall@**10
Базовый	0,93
Представленный	0,95

Таблица 1: Результаты эксперимента по поиску кандидатов.

Результаты экспериментов по сравнению фрагментов текста представлены в табл. 2. Представленный алгоритм показывает точность, сравнимую с точностью базового алгоритма и полноту, значительно превосходящую полноту базового алгоритма. Точность базового алгоритма объясняется тем, что данный алгоритм учитывает схожесть только почти-дубликатов текста.

Алгоритм	**Precision**	**Recall**	**F1**
Базовый	0,99	0,15	0,26
Представленный	0,93	0,80	0,85

Таблица 2: Результаты экспериментов по поиску схожих фрагментов текста.

На третьем эксперименте, учитывавшем качество представленного алгоритма в целом, были получены следующие показатели: Precision = 0.83, Recall = 0.79, $F1 = 0.80$.

7. Результаты экспериментов на реальной коллекции научных документов

Для апробации представленного алгоритма был проведен эксперимент по поиску переводных заимствований на коллекции документов из электронной библиотеки eLibrary.ru[7]. Данная библиотека содержит научные документы, входящие в Российский индекс научного цитирования (РИНЦ). Данный ресурс также содержит дополнительные метаданные для каждого документа: заголовок, авторов документа, язык документа и принадлежность к тематике, соответствующей Государственному рубрикатору научно-технической информации (ГРНТИ). Для апробации алгоритма в качестве проверяемых документов D_{susp} были подготовлены 2,5 миллиона документов на русском языке.

В качестве коллекции документов D использовались документы из английской версии Wikipedia, документы на английском языке из eLibrary.ru и статьи ресурсы arXiv.org[8]. Суммарное количество полученных документов: 7,6 миллионов.

[7]http://elibrary.ru
[8]http://arxiv.org

Таблица 3: Результаты экспериментов для коллекции документов on eLibrary.ru

Тип	**Количество**
Переводные заимствования	921
Другие заимствования	2462
Двуязычные статьи	788
Цитирование законов	1567
Ошибочные срабатывания	507
Другое	423
Всего	7689

В силу большого количества проверяемых документов D_{susp} для дальнейшего анализа рассматривались документы, содержащие значительное количество найденных заимствований. Была получена 21 тысяча документов со значительным количеством заимствований. Из них было проанализировано 7,6 тысяч документов, выбранных случайно. Основной целью эксперимента было детектирование кросс-языковых заимствований, когда заимствование было произведено из англоязычного документа в русскоязычный документ. В то же время, во время анализа полученных результатов, был выявлен ряд других срабатываний представленного алгоритма, которые были в дальнейшем разделены на несколько типов:

- Переводные заимствования — документ содержит заимствования, переведенные с английского языка, выданные за оригинальный текст.
- Другие заимствования — заимствования из русскоязычных ресурсов или заимствования, направление которого нельзя определить по датам документов.
- Двуязычные статьи — работы одного и того же автора на двух языках.
- Самоцитирование — цитирование автором его англоязычной работы.
- Цитирование законов — использование формулировок нормативных актов.
- Ошибочные срабатывания — ложно-положительные срабатывания представленного алгоритма.
- Другое — срабатывания, которые сложно отнести к какой-либо категории из-за нехватки метаданных или плохого качества текстов.

Результаты экспериментов представлены в табл. 3. Заметим, что было проанализировано только 36% всех срабатываний алгоритма, поэтому можно предварительно оценить количество документов с переводными заимствованиями по всей коллекции в 2,5 тысячи, что составляет 0.1% всех документов. Данную оценку можно существенно увеличить, проиндексировав большее количество англоязычных документов. Заметим также, что результаты были получены в автоматическом режиме и требуют дальнейшей экспертной верификации.

Распределение процента заимствований в проанализированных документах представлено на рис. 4. Средний процент заимствований составляет 20%.

Для анализа научных тематик, в которых переводные заимствования производятся наиболее часто были проанализированы документы, распределенные в тип *Переводные заимствования*. Около 70% анализированных документов были расклассифицированы по 10 научным рубрикам. Наибольшая часть документов была распределена между рубриками “Экономика. Народное хозяйство. Экономические науки” и “Право. Юридические науки”. Заметим, что распределение рубрик, распределенных в тип в тип *Двуязычные статьи* значительно отличается от данного распределения. Диаграммы 10 наибольших рубрик для данных типов срабатываний представлены на рис. 3.

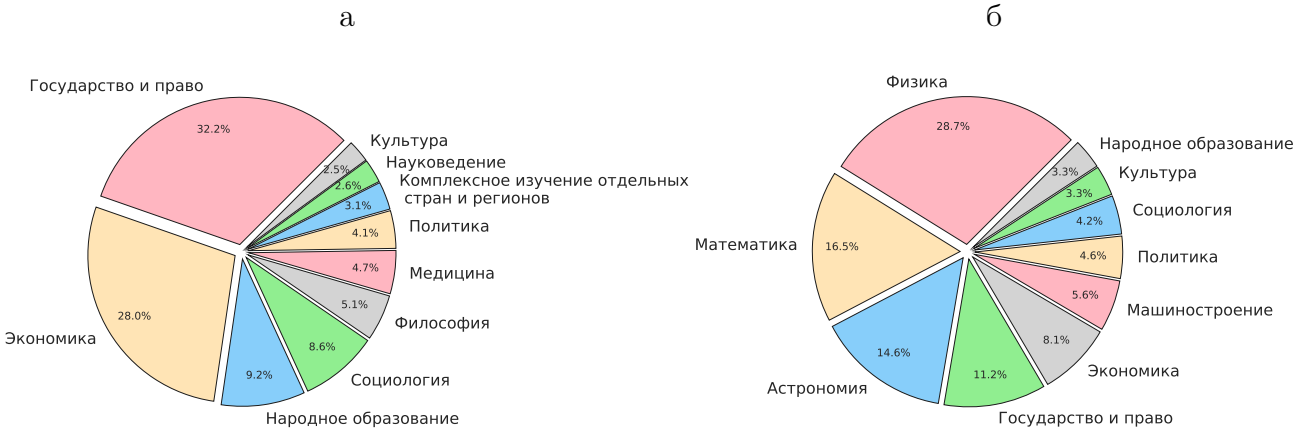


Рис. 3: Распределение рубрик ГРНТИ для типов: а) Переводные заимствования, б) Двуязычные статьи.

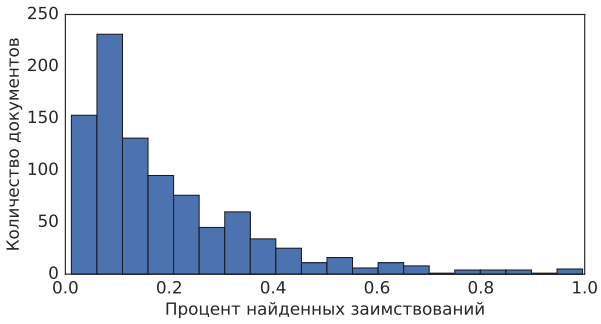


Рис. 4: Гистограмма распределения процента заимствованных текстов.

Анализ ложно-отрицательных срабатываний. Для анализа ложно-отрицательных срабатываний представленного алгоритма была проанализирована полнота нахождения двуязычных документов. Оценка полноты была проведена с помощью метаданных, полученных из eLibrary.ru в предположении. Анализ срабатываний алгоритма показал, что только 85% документов были найдены алгоритмом корректно. Заметим, что представленная оценка полноты является грубой, т.к. учитывает только полные переводы текстов.

Основная причина ложно-отрицательных срабатываний — низкое качество машинного перевода. Другой проблемой, значительно повлиявшей на качество нахож-

дения двуязычных статей, является используемый алгоритм поиска кандидатов, позволяющем находить только близкие по структуре заимствования. Кроме того, значительная часть часть проанализированных документов имела некорректную кодировку, что также повлияло на полноту поиска документов.

Анализ ложно-положительных срабатываний. Для анализа ложно-положительных срабатываний было проанализировано вручную 90 документов, определенных в тип *Ошибочные срабатывания*. Основная проблема ложно-положительных срабатываний состояла в некорректном векторном представлении предложений, содержащих именованные сущности, не встречаемые в обучающей выборке, а также содержащий слова, незнакомые модели машинного перевода. Также было замечено, что алгоритм сравнения документов часто находил общие фразы вида "Работа посвящена следующей проблеме: ..." и т.п. Несмотря на корректность данных срабатываний, общие фразы представленного вида встречаются в большом количестве документов, и потому не должны рассматриваться как кросс-языковые заимствования. Общий процент документов с ложно-положительными срабатываниями составил 7%.

8. Заключение

В работе был предложен алгоритм детектирования кросс-языковых заимствований. Для анализа качестве представленного алгоритма были проведены эксперименты на синтетических данных для пары языков "русский-английский". Качество алгоритма было также продемонстрировано на коллекции русскоязычных документов, входящих в Российский индекс научного цитирования (РИНЦ). В дальнейшем планируется развитие предложенного алгоритма: использование модели векторного представления предложений для задачи поиска кандидатов и улучшение качества отображения, ставящего в соответствие фразе вектор.

Авторы выражают свою благодарность Г.О. Еременко, ООО «Научная электронная библиотека» за предоставленные материалы.

СПИСОК ЛИТЕРАТУРЫ

1. Плагиат в работах студентов и аспирантов: проблема и методы противодействия / / Никитов А. В., Орчаков О. А., Чехович Ю. В. // Университетское управление: практика и анализ /. 2012. № 5. Pp. 61–68.

2. Plagiarism detection across distant language pairs / Alberto Barrón-Cedeno, Paolo Rosso, Eneko Agirre, Gorka Labaka // Proceedings of the 23rd International Conference on Computational Linguistics / Association for Computational Linguistics. 2010. Pp. 37–45.

3. *Franco-Salvador Marc, Rosso Paolo, Montes-y Gómez Manuel.* A systematic study of knowledge graph analysis for cross-language plagiarism detection // *Information Processing & Management.* 2016. Vol. 52, no. 4. Pp. 550–570.

4. *Hanane Ezzikouri, Erritali Mohammed, Oukessou Mohamed.* Semantic Similarity/Relatedness for Cross language plagiarism detection // Computer Graphics, Imaging and Visualization (CGiV), 2016 13th International Conference on / IEEE. 2016. Pp. 372–374.

5. Discovering text reuse in large collections of documents: A study of theses in history sciences / Anton S Khritankov, Pavel V Botov, Nikolay S Surovenko et al. // Artificial Intelligence and Natural Language and Information Extraction, Social Media and Web Search FRUCT Conference (AINL-ISMW FRUCT), 2015 / IEEE. 2015. Pp. 26–32.

6. *Grman Ján, Ravas Rudolf.* Improved Implementation for Finding Text Similarities in Large Collections of DataNotebook for PAN at CLEF 2011 // Notebook Papers of CLEF 2011 Labs and Workshops, 19-22 September, Amsterdam, The Netherlands / Ed. by Vivien Petras, Pamela Forner, Paul D. Clough. 2011. http://www.clef-initiative.eu/publication/working-notes.

7. *Grozea Cristian, Popescu Marius.* The Encoplot Similarity Measure for Automatic Detection of PlagiarismNotebook for PAN at CLEF 2011 // Notebook Papers of CLEF 2011 Labs and Workshops, 19-22 September, Amsterdam, The Netherlands / Ed. by Vivien Petras, Pamela Forner, Paul D. Clough. 2011. http://www.clef-initiative.eu/publication/working-notes.

8. Fuzzy Semantic Plagiarism Detection / Ahmed Hamza Osman, Naomie Salim, Yogan Jaya Kumar, Albaraa Abuobieda // International Conference on Advanced Machine Learning Technologies and Applications / Springer. 2012. Pp. 543–553.

9. *Vashchilin Serhii, Kushnir Halyna.* Comparison plagiarism search algorithms implementations // Advanced Information and Communication Technologies (AICT), 2017 2nd International Conference on / IEEE. 2017. Pp. 97–100.

10. Comparisons of keyphrase extraction methods in source retrieval of plagiarism detection / Hui Ning, Leilei Kong, Mingxing Wang et al. // Computer Science and Network Technology (ICCSNT), 2015 4th International Conference on / IEEE.

    Vol. 1. 2015. Pp. 661–664.

11. *Dutta Sandipan, Bhattacharjee Debotosh.* Plagiarism Detection by Identifying the Keywords // Computational Intelligence and Communication Networks (CICN), 2014 International Conference on / IEEE. 2014. Pp. 703–707.

12. Cross-language plagiarism detection over continuous-space-and knowledge graph-based representations of language / Marc Franco-Salvador, Parth Gupta, Paolo Rosso, Rafael E Banchs // *Knowledge-Based Systems.* 2016. Vol. 111. Pp. 87–99.

13. Using Word Embedding for Cross-Language Plagiarism Detection / Jérémy Ferrero, Frédéric Agnès, Laurent Besacier, Didier Schwab // EACL 2017. Vol. 2. 2017. Pp. 415–421.

14. Learning Paraphrastic Sentence Embeddings from Back-Translated Bitext / John Wieting, Jonathan Mallinson, Kevin Gimpel // EMNLP 2017. 2017. Pp. 274–285.

15. Moses: Open source toolkit for statistical machine translation / Philipp Koehn, Hieu Hoang, Alexandra Birch et al. // Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions / Association for Computational Linguistics. 2007. Pp. 177–180.

16. *Sutskever Ilya, Vinyals Oriol, Le Quoc V.* Sequence to Sequence Learning with Neural Networks // Proceedings of the 27th International Conference on Neural Information Processing Systems. NIPS'14. Cambridge, MA, USA: MIT Press, 2014. Pp. 3104–3112. http://dl.acm.org/citation.cfm?id=2969033.2969173.

17. Skip-thought vectors / Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov et al. // Advances in neural information processing systems. 2015. Pp. 3294–3302.

18. External and intrinsic plagiarism detection using a cross-lingual retrieval and segmentation system / Markus Muhr, Roman Kern, Mario Zechner, Michael Granitzer // Notebook Papers of CLEF 2010 LABs and Workshops. 2010.

19. A monolingual approach to detection of text reuse in Russian-English collection / Oleg Bakhteev, Rita Kuznetsova, Alexey Romanov, Anton Khritankov // Artificial Intelligence and Natural Language and Information Extraction, Social Media and Web Search FRUCT Conference (AINL-ISMW FRUCT), 2015 / IEEE. 2015. Pp. 3–10.

20. Plagiarism detection across distant language pairs / Alberto Barrón-Cedeno, Paolo Rosso, Eneko Agirre, Gorka Labaka // Proceedings of the 23rd International Conference on Computational Linguistics / Association for Computational Linguistics. 2010. Pp. 37–45.

21. *Ehsan Nava, Tompa Frank Wm., Shakery Azadeh.* Using a Dictionary and N-gram Alignment to Improve Fine-grained Cross-Language Plagiarism Detection // Proceedings of the 2016 ACM Symposium on Document Engineering.

    DocEng '16. New York, NY, USA: ACM, 2016. Pp. 59–68. http://doi.acm.org/10.1145/2960811.2960817.

22. *Alaa Zaid, Tiun Sabrina, Abdulameer Mohammedhasan.* Cross-language plagiarism of Arabic-English documents using linear logistic regression // *Journal of Theoretical and Applied Information Technology.* 2016. Vol. 83, no. 1. P. 20.

23. *Navigli Roberto, Ponzetto Simone Paolo.* BabelNet: Building a very large multilingual semantic network // Proceedings of the 48th annual meeting of the association for computational linguistics / Association for Computational Linguistics. 2010. Pp. 216–225.

24. *Miller George A.* WordNet: a lexical database for English // *Communications of the ACM.* 1995. Vol. 38, no. 11. Pp. 39–41.

25. Analysis on the Effect of Term-Document's Matrix to the Accuracy of Latent-Semantic-Analysis-Based Cross-Language Plagiarism Detection / Anak Agung Putri Ratna, F. Astha Ekadiyanto, Mardiyah et al. // Proceedings of the Fifth International Conference on Network, Communication and Computing.

ICNCC '16. New York, NY, USA: ACM, 2016. Pp. 78–82. http://doi.acm.org/10.1145/3033288.3033300.

26. Cross-Language Plagiarism Detection System Using Latent Semantic Analysis and Learning Vector Quantization / Anak Agung Putri Ratna, Prima Dewi Purnamasari, Boma Anantasatya Adhi et al. // *Algorithms*.

    2017. Vol. 10, no. 2. P. 69.

27. *Potthast Martin, Stein Benno, Anderka Maik.* A Wikipedia-based multilingual retrieval model // *Advances in Information Retrieval*.

    2008. Pp. 522–530.

28. *Landauer Thomas K, Dumais Susan.* Latent semantic analysis // *Scholarpedia*. 2008. Vol. 3, no. 11. P. 4356.

29. *Franco-Salvador Marc, Gupta Parth, Rosso Paolo.* Cross-language plagiarism detection using a multilingual semantic network // European Conference on Information Retrieval / Springer. 2013. Pp. 710–713.

30. *Le Quoc, Mikolov Tomas.* Distributed representations of sentences and documents // Proceedings of the 31st International Conference on Machine Learning (ICML-14). 2014. Pp. 1188–1196.

31. *Dai Andrew M, Olah Christopher, Le Quoc V.* Document embedding with paragraph vectors // *arXiv preprint arXiv:1507.07998*. 2015.

32. Off the Beaten Path: Let's Replace Term-Based Retrieval with k-NN Search / Leonid Boytsov, David Novak, Yury Malkov, Eric Nyberg // Proceedings of the 25th ACM International on Conference on Information and Knowledge Management / ACM. 2016. Pp. 1099–1108.

33. *Tai Kai Sheng, Socher Richard, Manning Christopher D.* Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks // *CoRR*. 2015. Vol. abs/1503.00075. http://arxiv.org/abs/1503.00075.

34. Learning Discriminative Projections for Text Similarity Measures / Wen-tau Yih, Kristina Toutanova, John C. Platt, Christopher Meek // Proceedings of the Fifteenth Conference on Computational Natural Language Learning. CoNLL '11. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011. Pp. 247–256. http://dl.acm.org/citation.cfm?id=2018936.2018965.

35. Deep Unordered Composition Rivals Syntactic Methods for Text Classification. / Mohit Iyyer, Varun Manjunatha, Jordan L. Boyd-Graber, Hal Daumé III // ACL (1). The Association for Computer Linguistics, 2015. Pp. 1681–1691. http://dblp.uni-trier.de/db/conf/acl/acl2015-1.html#IyyerMBD15.

36. Grounded Compositional Semantics for Finding and Describing Images with Sentences / Richard Socher, Andrej Karpathy, Quoc V. Le et al. // *TACL*. 2014. Vol. 2. Pp. 207–218. https://tacl2013.cs.columbia.edu/ojs/index.php/tacl/article/view/325.

37. Towards Universal Paraphrastic Sentence Embeddings / John Wieting, Mohit Bansal, Kevin Gimpel, Karen Livescu // *CoRR*. 2015. Vol. abs/1511.08198. http://arxiv.org/abs/1511.08198.

38. *Schmidhuber Jürgen*. Long short-term memory // *Neural computation*. 1997. Vol. 9, no. 8. Pp. 1735–1780.

39. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling / Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, Yoshua Bengio // *arXiv preprint arXiv:1412.3555*. 2014.

40. *Li Jiwei, Luong Minh-Thang, Jurafsky Dan*. A Hierarchical Neural Autoencoder for Paragraphs and Documents. // ACL (1). The Association for Computer Linguistics, 2015. Pp. 1106–1115. http://dblp.uni-trier.de/db/conf/acl/acl2015-1.html#LiLJ15.

41. An autoencoder approach to learning bilingual word representations / Sarath Chandar, Stanislas Lauly, Hugo Larochelle et al. // Advances in Neural Information Processing Systems. 2014. Pp. 1853–1861.

42. *Zhang Biao, Xiong Deyi, Su Jinsong*. BattRAE: Bidimensional Attention-Based Recursive Autoencoders for Learning Bilingual Phrase Embeddings // Proc. of AAAI. 2017.

43. Semi-supervised Recursive Autoencoders for Predicting Sentiment Distributions / Richard Socher, Jeffrey Pennington, Eric H. Huang et al. // Proceedings of the Conference on Empirical Methods in Natural Language Processing. EMNLP '11. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011. Pp. 151–161. http://dl.acm.org/citation.cfm?id=2145432.2145450.

44. *He Hua, Gimpel Kevin, Lin Jimmy J.* Multi-Perspective Sentence Similarity Modeling with Convolutional Neural Networks. // EMNLP / Ed. by Lluís Màrquez, Chris Callison-Burch, Jian Su et al. The Association for Computational Linguistics, 2015. Pp. 1576–1586.

45. Hashing for similarity search: A survey / Jingdong Wang, Heng Tao Shen, Jingkuan Song, Jianqiu Ji // *arXiv preprint arXiv:1408.2927*. 2014.

46. *Tiedemann Jörg*. News from OPUS - A Collection of Multilingual Parallel Corpora with Tools and Interfaces // Recent Advances in Natural Language Processing / Ed. by N. Nicolov, K. Bontcheva, G. Angelova, R. Mitkov. Borovets, Bulgaria: John Benjamins, Amsterdam/Philadelphia, 2009. Vol. V. Pp. 237–248.

47. *Federico Marcello, Bertoldi Nicola, Cettolo Mauro*. IRSTLM: an open source toolkit for handling large scale language models // INTERSPEECH. 2008.