*ВЕСТНИК ЮГОРСКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА*
2018 г. Выпуск 3 (50). С. 37–48

DOI: 10.17816/byusu2018037-48
УДК 004.85

Д. С. Ботов, Ю. Д. Кленин, И. Е. Николаев

# ИЗВЛЕЧЕНИЕ ИНФОРМАЦИИ С ИСПОЛЬЗОВАНИЕМ НЕЙРОСЕТЕВЫХ МОДЕЛЕЙ ЯЗЫКА НА ПРИМЕРЕ АНАЛИЗА ВАКАНСИЙ В СИСТЕМАХ ОНЛАЙН-РЕКРУТМЕНТА

Исследование выполняется в рамках проекта
«Интеллектуальная система формирования образовательных программ
на основе нейросетевых моделей естественного языка
с учетом потребностей цифровой экономики
Ханты-Мансийского автономного округа – Югры»,
который был поддержан грантом Российского фонда
фундаментальных исследований
№ 18-47-860013 р_а (договор № 18-47-860013\18).

*В статье рассматривается подход к извлечению информации с помощью онлайн-обучения на основе определения семантической близости векторов предложений и сущностей базы знаний с помощью нейросетевых моделей языка, обученных без учителя на большом текстовом корпусе предметной области. Приводится подробный обзор современных методов извлечения информации с учителем и без. Предложенный метод позволяет без трудоемкой процедуры разметки текстового корпуса и без применения подходов, основанных на правилах, достичь приемлемого качества в решении задачи анализа актуальных требований рынка труда. В рамках исследования профессиональные стандарты выступают в роли базы знаний предметной области с ограниченной лексикой. В основе подхода лежит определение семантической близости между векторными представлениями текстов, полученных с помощью различных нейросетевых моделей языка: усредненный word2vec, взвешенный по SIF усредненный word2vec, взвешенный по TF-IDF усредненный word2vec, paragraph2vec. В ходе эксперимента лучшее качество работы было показано моделью усредненного word2vec (CBOW).*

*Ключевые слова: машинное обучение, обработка естественного языка, нейросетевые модели языка, метод классификации, извлечение информации, распознавание именованных сущностей.*

D. S. Botov, J. D. Klenin, I. E. Nikolaev

# INFORMATION EXTRACTION USING NEURAL LANGUAGE MODELS FOR THE CASE OF ONLINE JOB LISTINGS ANALYSIS

*In this article we discuss the approach to information extraction (IE) using neural language models. We provide a detailed overview of modern IE methods: both supervised and unsupervised. The proposed method allows to achieve a high quality solution to the problem of analyzing the relevant labor market requirements without the need for a time-consuming labelling procedure. In this experiment, professional standards act as a knowledge base of the labor domain. Comparing the descriptions of work actions and requirements from professional standards with the elements of job listings, we extract four entity types. The approach is based on the classification of vector representations of texts, generated using various neural language models: averaged word2vec, SIF-weighted averaged word2vec, TF-IDF-weighted averaged word2vec, paragraph2vec. Experimentally, the best quality was shown by the averaged word2vec (CBOW) model.*

*Keywords: machine learning, natural language processing, neural language models, classification method, information extraction, named entity recognition.*

## Введение

Задача извлечения информации из текстов на естественном языке не теряет своей актуальности. Традиционными методами для данного направления задач являются подходы, основанные на правилах, а также методы машинного обучения с учителем на корпусах текстов, размеченных экспертами. Данные подходы показывают лучшее качество и для русского языка, что можно видеть на примере результатов соревнования по распознаванию именованных сущностей и извлечению фактов FactRuEval 2016 [28]. Однако не до конца исследованы возможности качественного решения задач извлечения информации в отдельных предметных областях с применением нейросетевых моделей, таких как word2vec, fasttext, paragraph2vec, а также других вариантов распределенных представлений слов, например, GloVe, обучаемых без учителя на больших текстовых корпусах. Хотя данные подходы к векторизации текстов показывают лучшие результаты в задачах определения семантической близости и разрешения лексической многозначности, в том числе и для русского языка [26, 27].

Целью исследования является экспериментальная оценка подхода к определению типа сущности, описываемой фрагментом текста на естественном языке, с помощью нейросетевых моделей, обученных без учителя на больших текстовых корпусах. В основе метода лежит гипотеза, что семантика типов сущностей, определенных в базе знаний (в рамках данного исследования составленной автоматически из ограниченного набора текстов профессиональных стандартов), соответствует семантике сущностей, используемых в описаниях вакансий работодателями в системах онлайн-рекрутмента (например, HeadHunter, Superjob), однако необходимо учесть разницу между лексикой профстандартов и вакансий.

Возможности применения данного подхода демонстрируются на примере задачи извлечения сущностей из текстов вакансий систем онлайн-рекрутмента путем сопоставления по семантической близости с текстами профессиональных стандартов для выявления актуальных требований рынка труда в ИТ-отрасли. Новизна исследования заключается в применении подходов на основе использования нейросетевых моделей языка в задаче извлечения сущностей из текста, не требующей трудоемкой ручной разметки корпуса для обучения классификаторов или написания сложной системы правил (без применения rule-based подходов).

## Обзор методов извлечения информации

Термин «извлечение информации» покрывает большой спектр задач обработки естественного языка. Можно выделить две основные составляющие процесса извлечения информации:

1) распознавание именованных сущностей;
2) извлечение отношений между сущностями.

## Распознавание сущностей

Распознавание сущностей (как правило, именованных, например, персон, организаций и локаций) связано с выделением отдельных фраз и предложений и определением их как упоминаний элемента того или иного типа.

Существуют следующие три подхода к распознаванию сущностей:

- газетиры – примитивные сборники различных вариантов упоминаний тех или иных сущностей, выступающие в роли словарей;
- rule-based – системы на основе правил, более широкие в своем обхвате, но ограниченные количеством заданных им указаний и шаблонов;
- алгоритмы машинного обучения – более сложные модели, способные выявлять сущности гибче, постепенно запоминая всё новые атрибуты таких элементов в процессе обучения.

На сегодняшний день базовым вариантом решения задачи распознавания сущностей является комбинация газетиров, базовых правил и Conditional random field (CRF). Последний является одним из классических алгоритмов машинного обучения.

Такой набор алгоритмов использовался, например, в качестве baseline в соревновании WNUT2015 [1]. Большинство участников использовали CRF, а также классические нейронные сети прямого распространения (Feedforward neural network, FFNN) и алгоритмы Маркова. Помимо самих текстов, многие участники также использовали значения векторных представлений слов: word embedding, использующие алгоритмы word2vec и GloVe. К сожалению, в сравнении с базовым показателем в 32 % участникам удалось добиться лишь результата в 56,4 %, что свидетельствует о наличии на сегодняшний день определенных трудностей в процессе извлечения сущностей.

В работе [12] авторы предлагают алгоритм автоматического построения газетиров на основе WordNet и Wikipedia путем выявления типа сущности через передвижение вверх по иерархии гиперонимов. Метод показывает достаточно слабые результаты для таких типов именованных сущностей, как персона и организация, но работает лучше для географических локаций. Кроме того, он ограничен данными, имеющимися в упомянутых системах. Насколько нам известно, данный метод развития не получил.

Системы на основе правил считаются на сегодняшний день достаточно примитивными, пригодными лишь для автоматизации процессов извлечения информации, которая и так уже достаточно хорошо структурирована. Основным минусом систем на основе правил является их ограниченность – для каждого нового раздела знаний требуется разработка своего фонда правил, способных учесть специфику текстов этой области, что требует привлечения большого количества человеческих ресурсов. В то же время качество работы более автоматизированных систем на основе алгоритмов машинного обучения возросло достаточно, чтобы конкурировать с самыми лучшими системами на базе правил. В [2] авторами была заявлена возможность разработки rule-based системы, способной сравниться по качеству с алгоритмами machine learning, если предварительно потратить 8 человеко-недель на разработку правил под конкретную предметную область.

Говоря об алгоритмах машинного обучения, стоит разделять алгоритмы, действующие с учителем, – обученные на достаточном объеме заранее размеченных вручную примеров, и без учителя – обучающиеся распознавать сущности, пользуясь исключительно информацией, предоставленной в обрабатываемых данных и некоторыми заранее известными эвристиками. Алгоритмы с учителем имеют недостаток сродни системам на основе правил: для их обучения требуется достаточно трудоемкий процесс подготовки обучающих данных.

Среди алгоритмов машинного обучения с учителем большинство классических методов сводит задачу распознавания сущностей к разметке последовательностей и их последующей поэлементной классификации. Из более конкретных примеров можно выделить уже упомянутые выше CRF. CRF является одной из наиболее популярных моделей поиска именованных сущностей, определяя тэги на основе атрибутов, но с учетом как текущего слова, так и предыдущих, и последующих слов в тексте. Так, этот алгоритм лежит в основе ряда популярных разметчиков последовательностей [3–5].

Также встречаются алгоритмы разметки последовательностей на основе максимальной энтропии [6], предсказывающей метку элемента последовательности на основе вероятностей возникновения тех или иных атрибутов слова и его предшественников, и марковских моделей [7], воспринимающих разметку текста как марковский процесс, где состояния – это искомые классы, а вероятности меток текущего элемента определяются предыдущим состоянием процесса.

Более сложные алгоритмы классификации последовательностей могут опираться на сложные нейросетевые модели, такие как LSTM, получившей популярность в работе с текстовыми данными ввиду способности к учету истории пропущенных через нее последовательностей. Примеры использования таких моделей можно найти в [13], где использование двунаправленных LSTM-сетей позволяет одновременно учитывать атрибуты как предыдущих, так и после-

дующих слов в предложении при назначении тэга сущности, и в [14], где авторы сравнивают производительность однонаправленных и двунаправленных LSTM с применением CRF на выходе сети для учета полученных тэгов соседних слов и повышения качества.

В отличие от алгоритмов с учителем, алгоритмы без учителя зачастую осуществляют выявление сущностей в тексте на основе поиска схожих слов в документе, в попытке выявить именованные сущности в общие группы, исходя из контекста.

Примером данного подхода является [8], в котором авторы применяют Word2vec для генерации кластеров слов с близкими контекстами. Такой подход показывает лучшие результаты в сравнении с классическим CRF для языков с низким объемом размеченных корпусов (например, Бенгальский язык). Другим примером является использование Брауновских кластеров – организация слов в документе в иерархические кластера на основе вероятностей их распределения [9].

Однако зачастую алгоритмы с учителем и без учителя используются в тандеме. Например, в работе [10] автор предлагает использование упомянутой выше модели word2vec, вектора слов которой используются в качестве одного из атрибутов в процессе классификации именованных сущностей. Авторы [11] производят сопоставление качества различных вариантов использования word2vec и Брауновских кластеров в качестве атрибутов для CRF-классификатора для поиска сущностей в медицинских текстах.

**Извлечение отношений**

Извлечение отношений между найденными в тексте сущностями является следующим логическим шагом в извлечении структурированной информации из простого неструктурированного текста. В литературе существует несколько подходов к решению данной задачи.

Одним из этих подходов является подход на основе классификации возможных кандидатов-пар сущностей. Так, в работе [15] авторы представляют свой алгоритм классификации бинарных отношений на основе множества различных групп атрибутов – статистических (частотных), языковых, основанных на уже имеющихся знаниях, – использующий SVM в качестве классификатора типов отношений. Другим примером является алгоритм на основе сверточной нейронной сети в работе [16]. Здесь CNN используется для объединения атрибутов отдельных слов и получения атрибутов всего предложения, которые затем используются все вместе для обучения softmax-классификатора типов отношений.

Другая группа подходов основывается на использовании ядер функций (kernels) для автоматизированного выявления шаблонов определенных видов отношений. Например, авторы [17] рассматривают применение tree-kernel для построения синтаксических структур и определения близостей между ними, используя наличие в них общих поддеревьев. В другом примере [18] авторы предлагают метод на основе полиномиального ядра для автоматического поиска слов «взаимодействия», участвующих в шаблонах отношений.

Третья группа подходов трактует задачу извлечения отношений как задачу генерации. Более конкретно, используя исходный неструктурированный текст, такие подходы производят генерацию структурированного представления информации, содержащейся в тексте. Такая генерация, как правило, попадает под термин sequence2sequence или seq2seq – «последовательность в последовательность».

В [19] вариация seq2seq на основе двунаправленных рекуррентных сетей (BiRNN) используется для извлечения троек отношений из неструктурированного текста. Данная нейросеть использует вектора уверенности, чтобы определить, относится ли та или иная тройка к конкретному типу отношений, а также встречаются ли упоминания сущностей во входной последовательности. Данный метод ограничен необходимостью задания точного количества типов отношений заранее, что уменьшает его гибкость при смене домена. Авторы [20] рассматривают возможность применения seq2seq для автоматического извлечения фактов из текстов статей Wikipedia в формате ее infobox-элементов. В данном случае авторами используется отдельный генератор для каждого типа элемента.

В работе [21] авторы предлагают использование CNN-LSTM энкодера-декодера для преобразования входного предложения в набор всех присутствующих в нем отношений, в порядке убывания содержательности информации.

Основной недостаток данного подхода – потребность в достаточном размере учебного корпуса, необходимого для качественного обучения seq2seq. Кроме того, данный корпус будет больше, нежели корпуса для обучения обычного классификатора.

### Метод определения типа сущностей

В рамках исследования предлагается поставить задачу определения типа сущности для фрагмента текста вакансии путем соотнесения с элементами профессиональных стандартов.

### Концептуальная модель

Рисунок 1 иллюстрирует концептуальную модель предметной области, описывающей соответствие элементов описания вакансий и элементов профессиональных стандартов.

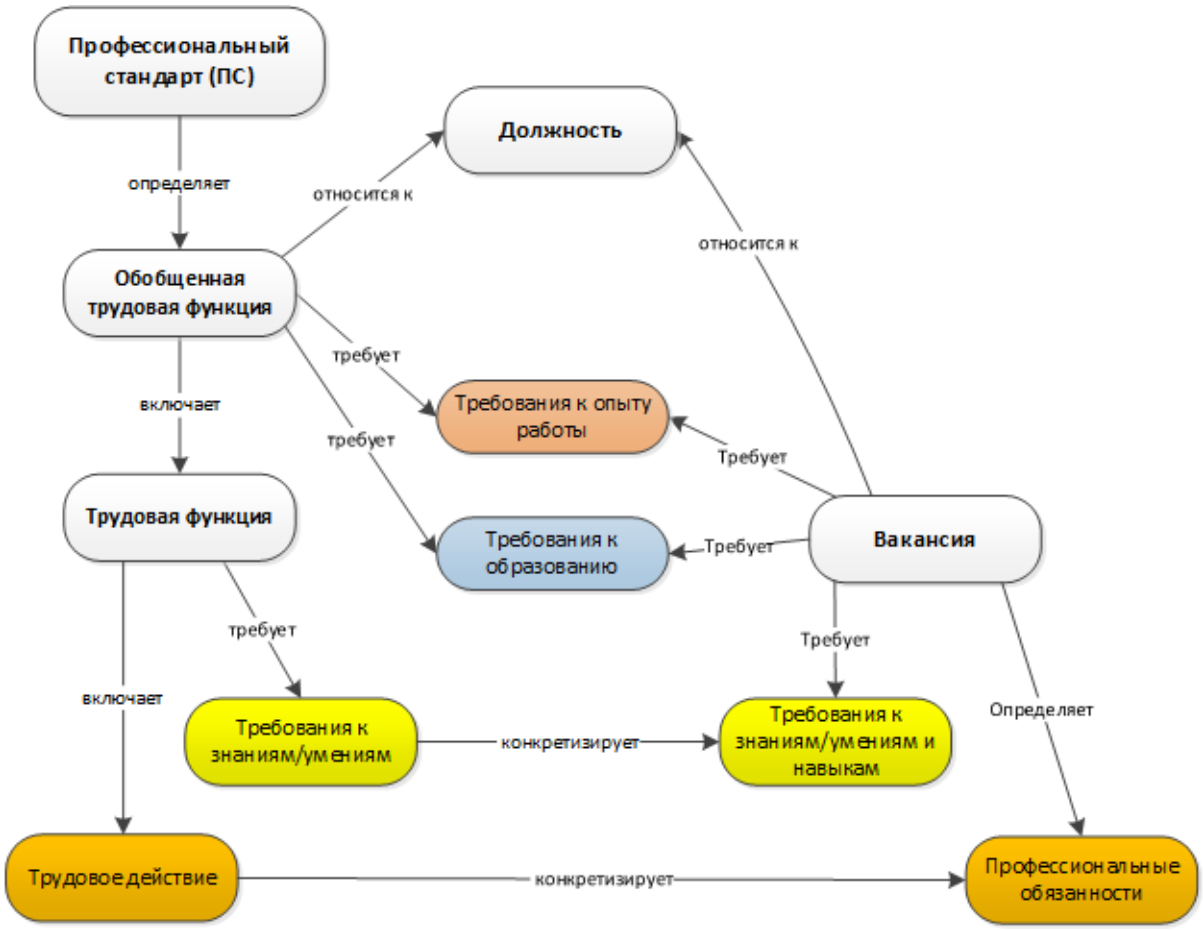


Рисунок 1 – Концептуальная модель связи сущностей между профессиональными стандартами и вакансиями

В рамках данной работы предлагается определять следующие распространенные в описаниях вакансий типы сущностей:

- трудовые действия (обязанности);
- требования к образованию;
- требования к знаниям/умениям;
- требования к опыту работы.

**Алгоритм определения типа сущности**

На рисунке 2 представлена общая схема алгоритма определения типа сущности на основе определения ближайших по семантике элементов текстов стандартов для каждого из элементов вакансии. При этом семантическая близость текстов в данном случае определяется на основании близости векторных представлений текстов, сгенерированных нейросетевой моделью языка, обученной на большом корпусе текстов вакансий и профстандартов.

Векторная близость, как правило, определяется на основе косинусной меры угла между направлениями двух векторов: чем ближе вектора друг к другу, тем меньше угол между их направлениями, а значит ближе к 1 его косинус.

Для каждого элемента текста вакансии подход сводится к следующим этапам:

– определение ближайших элементов стандартов;
– голосование классов этих «соседей» для определения класса элемента вакансии.

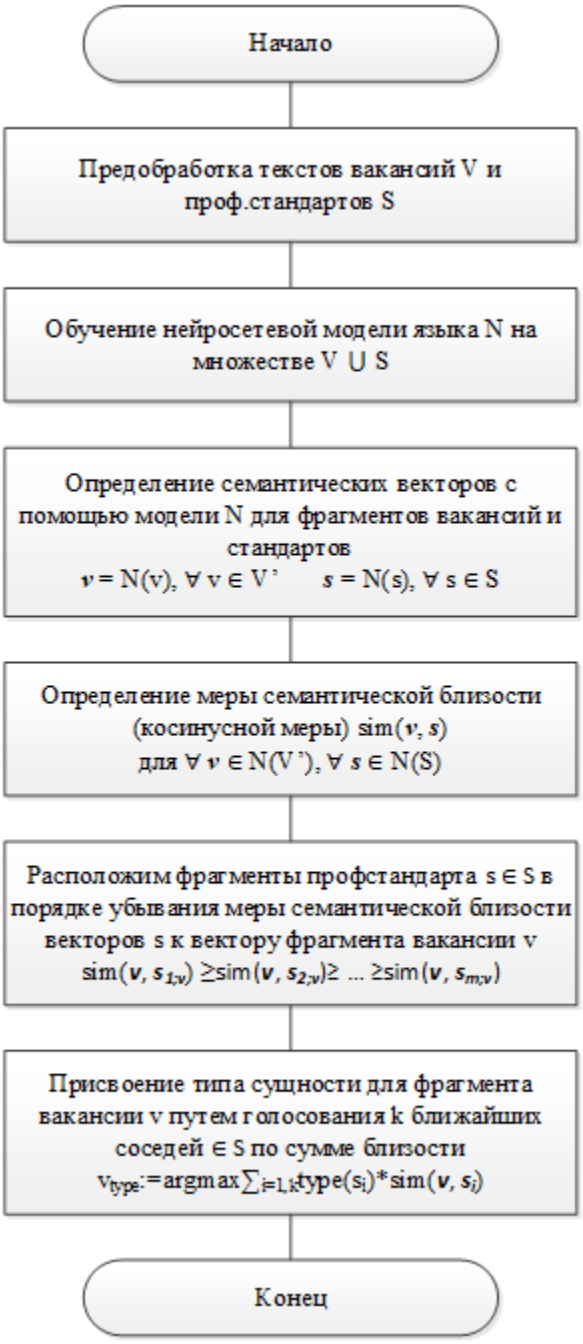


Рисунок 2 – Алгоритм определения типа сущности для фрагмента вакансии на основе голосования k-ближайших соседей из фрагментов стандартов

Таким образом, основным этапом предлагаемого подхода является векторизация текстового документа. В эксперименте осуществляется сравнение ряда различных известных алгоритмов векторных представлений: усредненный word2vec, взвешенный по TF-IDF усредненный word2vec, взвешенный по SIF усредненный word2vec, paragraph2vec. Эти алгоритмы показывают высокие результаты в задачах определения семантической близости, в том числе для русского языка, например, в рамках соревнований по определению семантической близости и по разрешению лексической многозначности, проводимых в рамках семинара RUSSE на конференции «ДИАЛОГ»[26, 27], а также в задачах кластеризации текстов [29].

## Нейросетевые модели языка

### Word2Vec

Нейросетевой подход моделирования языка был предложен командой исследователей Google под руководством T. Mikolov [22]. Данный подход представлен в виде двух вариаций нейросетевой архитектуры, содержащей единственный скрытый слой. Итоговая модель, полагаясь на дистрибутивную гипотезу (языковые единицы со схожими распределениями имеют схожий смысл), обучается сопоставлять слова и контексты их употребления. Обучение проходит без помощи учителя, используя лишь неразмеченные тексты, выдавая на выходе набор векторов заданной размерности для любого слова, встреченного в процессе обучения. При этом получившиеся вектора отражают близость этих слов: более близкие слова имеют более близкие вектора и наоборот.

Положительными характеристиками данной модели являются низкая разреженность итоговых векторов, возможность задания их размерности, а также скорость работы (в сравнении с более сложными моделями, дающими схожий уровень качества). Основным недостатком же является невозможность интерпретации значений координат некоторого вектора.

Для получения векторного представления целого текста необходимо объединить векторные представления отдельных слов, что осуществляется, как правило, с помощью взятия среднего значения векторов.

### Paragraph2Vec

Развивая идею word2vec, T. Mikolov вскоре предложил нейросетевую модель векторизации целых документов [23], называемую paragraph2vec или doc2vec. Эта модель имеет архитектуру, аналогичную word2vec, с той лишь разницей, что, помимо контекстных слов, модель также учитывает контекстный документ, заучивая в процессе обучения и его векторное представление. В результате paragraph2vec способна возвращать вектора целых текстов, имеющих сходное качество с векторами отдельных слов, в word2vec. При этом для ранее не встреченных документов вектор может быть сгенерирован на основе входящих в документ слов.

Таким образом, используя paragraph2vec, можно получать векторные представления текстов без каких-либо дополнительных действий.

### TF-IDF

Ставшая классическим алгоритмом в NLP, TF-IDF является простой в понимании и вычислении схемой взвешивания слов в документах. TF-IDF – это комбинация двух более простых весов для слова: tf (term frequency) *частота слова* и idf (inverse document frequency) *обратная частота документа*. TF – это простейшая частотная характеристика слова в корпусе документов, отражающая частоту его употребления в документах данного набора. Эвристика TF связана с предположением, что чем чаще слово употребляется, тем более оно важно. IDF – это чуть более сложная частотная характеристика, показывающая, насколько то или иное слово значимо для различения между текстами анализируемого корпуса. Данная мера веса пытается скорректировать недочет TF, из-за которого возрастает вес часто употребляемых, но мало значимых служебных слов. Для этого IDF обратно пропорциональна количеству документов, в которых то или иное слово встречается, придавая больший вес тем словам, которые встречаются лишь в отдельных документах, предполагая, что эти слова наиболее качественно описывают такой документ.

Векторные представления слов при использовании TF-IDF являются one-hot векторами, содержащими лишь одно значение, отличное от 0, равное TF-IDF весу этого слова. При этом размерность такого вектора равна количеству уникальных слов в том или ином корпусе. В таком случае для получения векторных представлений документа целиком создается аналогичный вектор, в котором вместо 0 подставляются TF-IDF веса всех встречаемых в нем слов.

Главным недостатком таких векторов является их чрезвычайная разреженность: в коллекции документов могут использоваться все слова того или иного языка (свыше 1 миллиона), в то время как в каждом из документов может употребляться лишь небольшая их часть (как правило, порядка 5–10 тысяч). Преимуществами же TF-IDF являются простота ее расчета и абсолютная прозрачность в интерпретации значений векторов.

Однако веса TF-IDF можно использовать как модификаторы для других векторных представлений. В таком случае TF-IDF вес того или иного слова является некоторым лексическим фильтром, определяющим влияние этого слова на итоговый вектор текста, снижая или повышая его вклад на основе «важности» этого слова.

В наших экспериментах мы используем TF-IDF-взвешивание для повышения качества усредненного word2vec.

**Smoothed Inverse Frequency (SIF)**

Другая форма векторизации текстов, основанная на взвешивании векторных представлений слов, была предложена учеными университета Принстона [24]. SIF (smoothed inverse frequency), или сглаженная обратная частота, – это алгоритм векторизации, состоящий из двух этапов.

Первый этап – вычисление весов слов. Изначально генеративная модель определяет вероятности генерации того или иного слова в текущий момент времени с учетом текущего контекста этого слова. Затем веса каждого слова определяются как:

$$weight = a/(a + p(w)),$$

где $a$ – это параметр, а $p(w)$ – это вероятность генерации слова $w$.

Результирующие веса используются для взвешивания исходных векторов слов.

Второй этап – удаление общих компонент. Данное действие схоже в мотивации с IDF: часто употребляемые слова и пары слов получают большие вектора, которые вызывают аномалию в векторах документов, полученных через усреднение векторов слов. А именно вклад векторов этих слов вызывает рост проекции вектора документа на направления, не имеющие никакого смысла. Для борьбы с этим влиянием авторы предлагают произвести удаление проекции вектора документа на первую главную компоненту.

В результате модель генерирует взвешенные усредненные вектора для документов, весьма простые в вычислении, но более эффективные, чем многие современные бейзлайны.

## Эксперимент

Далее рассматриваются отдельные аспекты эксперимента с оценкой качества реализуемого подхода на представительном текстовом корпусе.

### Характеристики текстовых корпусов

Для обучения нейросетевых моделей был подготовлен большой корпус из текстов вакансий в области информационных технологий с платформ онлайн-рекрутмента headhunter и superjob за последние 5 лет.

Для оценки качества моделей были подготовлены два корпуса:

1. Корпус профессиональных стандартов (утвержденных Минтруда РФ) по профессиям: «программист», «администратор баз данных», «системный администратор информационно-коммуникационных систем», «специалист по тестированию в области информационных технологий».

2. Корпус из фрагментов 102 вакансий соответствующих профессий, в котором 4-мя экспертами были размечены следующие типы сущностей (каждый фрагмент описывает только одну сущность):

- трудовые действия (обязанности): 576 примеров;
- требования к образованию: 40 примеров;
- требования к знаниям/умениям: 545 примеров;
- требования к опыту работы: 53 примера.

Подробные характеристики корпусов текстов представлены в таблице 1.

### Параметры обучения нейросетевых моделей

Для обучения нейросетевых языковых моделей использовалась реализация из библиотеки gensim [25]. В таблице 2 представлены параметры обучения этих моделей.

Таблица 1 – Характеристики текстовых корпусов, используемых в ходе экспериментов

| Корпус | Число документов | Число фрагментов (предложений) | Число токенов | Число уник. токенов (словарь) |
|---|---|---|---|---|
| Большой корпус вакансий | 653 тыс. | 1385 тыс. | 130 млн. | 200 тыс. |
| Корпус проф. стандартов | 4 стандарта | 1107 | 13 тыс. | 1,1 тыс. |
| Тестовый корпус вакансий | 102 | 1214 | 8,8 тыс. | 1,6 тыс. |

Таблица 2 – Параметры обучения нейросетевых моделей языка

| Модель | Архитектура | Размерность | Min. частота встречи слова | Эпохи |
|---|---|---|---|---|
| Paragraph2vec | PV-DM | 200 | 3 | 5 |
| Paragraph2vec | PV-DBOW | 200 | 3 | 5 |
| Word2vec | skip-gram | 300 | 3 | 5 |
| Word2vec | CBOW | 300 | 3 | 5 |

### Предобработка текста

Перед обучением моделей исходные тексты обрабатываются по следующим принципам:

1) многострочные тексты объединяются в одну строку;
2) тексты очищаются от всех символов, не являющихся буквами, цифрами, знаками пробела или некоторыми спецсимволами;
3) каждый токен подвергается морфологическому анализу и приводится к нормальной форме (если это возможно);
4) для нормализованных токенов производится добавление пометки части речи;
5) производится удаление служебных частей речи (союзов, предлогов и местоимений).

### Результаты эксперимента

Как следует из приведенных в таблице 3 результатов, применение различных модификаций взвешивания к усредненному вектору word2vec не привело к повышению качества решения задачи определения типа сущности. Также в таблице приведены значения k – числа ближайших соседей, при котором были получены лучшие результаты для каждой из моделей.

Таблица 3 – Результаты сравнения различных нейросетевых моделей языка в определении типа сущности по k-ближайшим соседям

| **Модель** | **Precision** | **Recall** | **F1 (micro)** | **k (число ближайших соседей)** |
|---|---|---|---|---|
| Doc2vec (DBOW) | 0.56 | 0.53 | 0.51 | 14 |
| Doc2vec (DM) | 0.47 | 0.49 | 0.47 | 14 |
| Avr. Word2Vec (skip-gram) | 0.69 | 0.67 | 0.68 | 13 |
| Avr. Word2Vec (CBOW) | 0.73 | 0.72 | 0.72 | 14 |
| TFIDF+Word2Vec | 0.64 | 0.62 | 0.62 | 14 |
| SIF+Word2Vec | 0.60 | 0.58 | 0.57 | 9 |

Это объясняется уже упомянутой выше разницей лексики между двумя корпусами: корпусом профессиональных стандартов и корпусом вакансий. Классические схемы взвешивания (TF-IDF, SIF) оказались не способны адаптироваться и качественно рассчитать инверсные частоты для терминов из словаря при отображении элементов вакансий на пространство элементов профессиональных стандартов.

Значительно хуже себя показал paragraph2vec, что можно объяснить низким качеством применения обученных моделей архитектур PV-DBOW и PV-DM к коротким текстам фрагментов вакансий (10–15 слов в среднем), а также малым числом примеров текстов стандартов в сравнении с числом текстов вакансий для качественного обучения контекстов документов.

Таблица 4 – Результаты распознавания типов сущностей по каждому из классов для лучшей модели Avr. Word2Vec (CBOW)

| **Тип сущности** | **Precision** | **Recall** | **F1 (micro)** |
|---|---|---|---|
| Трудовое действие (обязанность) | 0.78 | 0.70 | 0.74 |
| Требования к образованию | 0.97 | 0.88 | 0.92 |
| Требования к знаниям/умениям | 0.68 | 0.71 | 0.70 |
| Требования к опыту работы | 0.50 | 0.87 | 0.63 |
| **Итого (micro)** | **0.73** | **0.72** | **0.72** |
| **Итого (macro)** | **0.73** | **0.79** | **0.75** |

Однако следует отметить, что даже несмотря на полное отсутствие разметки вакансий в учебных данных, отображение документов одного типа на пространство документов другого типа на основе определения семантической близости с помощью нейросетевых моделей языка путем онлайн-обучения по методу ближайших соседей уже дает качество определения типа сущности в 0.72 по F-мере.

## Заключение

В данной работе был предложен и экспериментально исследован метод определения типа сущности при извлечении информации из текстов путем расчета семантической близости векторов, полученных с помощью нейросетевых моделей языка, и определения ближайших соседей-сущностей из автоматически построенной базы знаний. Применимость метода определения четырех типов сущностей из текстов вакансий продемонстрирована на представительном текстовом корпусе вакансий и профессиональных стандартов. В ходе эксперимента определена лучшая нейросетевая модель – усредненный word2vec, обученная по алгоритму

CBOW, которая на определении четырех типов сущностей показывает качество micro-F1: 0.72 и macro-F1: 0.75. Основная доля ошибок связана со спецификой формулировок обязанностей и требований в вакансиях, когда работодатели смешивают описание трудовых действий с требованиями к практическому опыту применения трудовых действий (навыков).

Метод имеет преимущества в низкой трудоемкости подготовки текстового корпуса в сравнении с традиционными методами обучения с учителем и методами, основанными на правилах. Также в ходе эксперимента показано преимущество использования векторов модели word2vec без схем взвешивания TF-IDF или SIF в условиях ограниченной лексики текстов из базы знаний, автоматически сгенерированной из профессиональных стандартов.

**Литература**

1. Shared tasks of the 2015 workshop on noisy user-generated text: Twitter lexical normalization and named entity recognition [Text] / T. Baldwin, de M.-C. Marneffe, B. Han [et al.] // In Proceedings of the Workshop on Noisy User-Generated Text. – Beijing, China, 2015. – P. 126–135.

2. Domain adaptation of rule-based annotators for named-entity recognition tasks [Text] / L. Chiticariu, R. Krishnamurthy, Y. Li [et al.] // Proceedings of the 2010 conference on empirical methods in natural language processing, Association for Computational Linguistics. – San Jose, USA, 2010. – P. 1002–1012.

3. Finkel, J. R. Incorporating non-local information into information extraction systems by gibbs sampling [Text] / J. R. Finkel, T. Grenager, C. Manning // Proceedings of the 43rd annual meeting on association for computational linguistics, Association for Computational Linguistics. – Stanford, USA, 2005. – P. 363–370.

4. Kudo, T. CRF++: Yet another CRF toolkit [Electronic resource] / T. Kudo // GitHub. – URL: https://github.com/taku910/crfpp.

5. Seker, G. A. Extending a CRF-based named entity recognition model for Turkish well formed text and user generated content [Text] / G. A. Seker, G. Eryigit // Semantic Web 8, IOS Press. – 2017. – № 5. – P. 625–642.

6. Bikel, D. M. An algorithm that learns what's in a name [Text] / D. M. Bikel, R. Schwartz, R. M. Weischedel // Machine learning 34. – 1999. – № 1–3. – P. 211–231.

7. Curran, J. R. Language independent NER using a maximum entropy tagger [Text] / J. R. Curran, S. Clark // Proceedings of the seventh conference on Natural language learning at HLT-NAACL. – 2003. – Vol. 4. – P. 164–167.

8. Das, A. Named entity recognition with word embeddings and wikipedia categories for a low-resource language [Text] / A. Das, D. Ganguly, U. Garain // ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP). – USA, New York. – 2017. – Vol. 16, Issue 3. – P. 19–25.

9. Class-based n-gram models of natural language [Text] / P. F. Brown, P. V. Desouza, R. L. Mercer [et al.] // Computational linguistics 18. – 1992. – № 4. – P. 467–479.

10. Siencnik, S. K. Adapting word2vec to named entity recognition [Text] / S. K. Siencnik // Proceedings of the 20th Nordic Conference of Computational Linguistics. – Sweden, 2015. – № 109. – P. 239–243.

11. Wu, Y. A study of neural word embeddings for named entity recognition in clinical text [Text] / Y. Wu, J. Xu, M. Jiang, Y. Zhang, H. Xu // AMIA Annual Symposium Proceedings, American Medical Informatics Association. – USA, San Francisco. – 2015. – Vol. 2015. – P. 1326–1333.

12. Toral, A. A proposal to automatically build and maintain gazetteers for Named Entity Recognition by using Wikipedia [Text] / A. Toral, R. Munoz // Proceedings of the Workshop on NEW TEXT Wikis and blogs and other dynamic text sources. – Italy, Trento. – 2006. – Vol. 1. – P. 56–61.

13. Chiu, J. P.-C. Named entity recognition with bidirectional LSTM-CNNs [Text] / J. P.-C. Chiu, E. Nichols // Transactions of the Association for Computational Linguistics. – 2016. – Vol. 4. – P. 357–370.

14. Huang, Z. Bidirectional LSTM-CRF models for sequence tagging [Text] / Z. Huang, W. Xu, K. Yu // arXiv preprint arXiv:1508.01991. – 2015.

15. RELigator: chemical-disease relation extraction using prior knowledge and textual information [Text] / E. Pons, B. F. H. Becker, S. A. Akhondi [et al.] // Proceedings of the Fifth BioCreative Challenge Evaluation Workshop. – Spain, Sevilla. – 2015. – Vol. 1. – P. 247–253.

16. Relation classification via convolutional deep neural network [Text] / D. Zeng, K. Liu, S. Lai [et al.] // Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers. – Ireland, Dublin. – 2014. – Vol. 1. – P. 2335–2344.

17. Plank, B. Embedding semantic similarity in tree kernels for domain adaptation of relation extraction [Text] / B. Plank, A. Moschitti // Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics. Volume 1: Long Papers. – Bulgaria, Sofia. – 2013. – Vol. 1. – P. 1498–1507.

18. Quan, C. An unsupervised text mining method for relation extraction from biomedical literature [Text] / C. Quan, M. Wang, F. Ren // PloS one – 2014. – Vol. 9, Issue 7. – P. 1–8.

19. Extracting Relational Facts by an End-to-End Neural Model with Copy Mechanism [Text] / X. Zeng, D. Zeng, S. He [et al.] // Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics. – Australia, Melbourne. – 2018. – Vol. 1. – P. 506–514.

20. Xu, B. CN-DBpedia: A never-ending Chinese knowledge extraction system [Text] / B. Xu, Y. Xu, J. Liang [et al.] // In International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, France, Arras : Springer International Publishing. – 2017. – Vol. 1, Part II, LNAI 10351 – P. 428–438.

21. Exploring Encoder-Decoder Model for Distant Supervised Relation Extraction [Text] / S. Su, N. Jia, X. Cheng [et al.] // Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18). – Sweden, Stockholm. – 2018. – Vol. 1. – P. 4389–4395.

22. Distributed representations of words and phrases and their compositionality [Text] / T. Mikolov, I. Sutskever, K. Chen [et al.] // In Advances in neural information processing systems. – 2013. – Vol. 1. – P. 3111–3119.

23. Le, Q. Distributed representations of sentences and documents [Text] / Q. Le, T. Mikolov // International Conference on Machine Learning. – China, Beijing. – 2014. – Vol. 32. – P.1188–1196.

24. Arora, S. A simple but tough-to-beat baseline for sentence embeddings [Text] / S. Arora, Y. Liang, T. Ma // International Conference on Learning Representations (ICLR). – France, Toulon. – 2017. – Vol. 1. – P. 1–16.

25. Rehurek, R. Software framework for topic modelling with large corpora [Text] / R. Rehurek, P. Sojka // Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks. – Malta, Valletta. – 2010. – Vol. 1. – P. 46–50.

26. RUSSE'2018: a Shared Task on Word Sense Induction for the Russian Language [Text] / A. Panchenko, A. Lopukhina, D. Ustalov [et al.] // Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference «Dialogue 2018». – Russia, Moscow. – 2018. – Vol. 1. – P. 547–564.

27. RUSSE: The First Workshop on Russian Semantic Similarity [Text] / A. Panchenko, N. V. Loukachevitch, D. Ustalov [et al.] // Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference «Dialogue 2015». – Russia, Moscow. – 2015. – Vol. 2. – P. 89–105.

28. FactRuEval 2016: Evaluation of Named Entity Recognition and Fact Extraction Systems for Russian [Text] / V. V. Bocharov, S. V. Alexeeva, A. A. Bodrova [et al.] // Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference «Dialogue 2016». – Russia, Moscow. – 2016. – Vol. 1. – P. 702–720.

29. Пархоменко, П. А. Обзор и экспериментальное сравнение методов кластеризации текстов [Текст] / П. А. Пархоменко, А. А. Григорьев, Н. А. Астраханцев // Труды ИСП РАН. – 2017. – Т. 29, Вып. 2. – С. 161–200.