

# МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

## РАЗРАБОТКА ПРОГРАММЫ «БАЗЫ ЗНАНИЙ ТОГУ» С ИСПОЛЬЗОВАНИЕМ ПОЛНОТЕКСТОВОГО ПОИСКА С ПОМОЩЬЮ ЛЕКСЕМ ЕСТЕСТВЕННОГО ЯЗЫКА

Забавин Алексей Сергеевич  
Тихоокеанский Государственный Университет  
Г. Хабаровск  
2025

## Предмет работы

Предметом исследования является разработка вопросно-ответной системы базы знаний ТОГУ. Изучение качества поиска — при простом поиске по вхождению текста, при индексировании на основе «частотной важности» слов в документе и полнотекстовом поиске по нему.

А также использование оптимизаций поискового запроса на основе семантической близости и синтаксической важности членов предложения в тексте документа

## **Объектами исследования являются:**

1. хранение информации для QA-системы в базе данных позволяющее решать задачу полнотекстового поиска в ней
2. частотный алгоритм ранжирования результатов поиска в коллекции документов;
3. методы лексического, синтаксического и семантического анализа текста.

## Задача полнотекстового поиска

Полнотекстовый поиск предназначен для поиска и ранжирования текстовых данных на основе ключевых слов или фраз, встречающихся в текстовых полях базы данных где стандартные механизмы вроде оператора LIKE недостаточны.

Поиск должен учитывать различные формы слов.

Важным аспектом является не только нахождение документов, но и их сортировка по релевантности. Стандартные SQL-запросы не обладают встроенной поддержкой ранжирования результатов на основе того, насколько близки слова запроса к друг другу в документе или как часто они встречаются.

# Разработанная система



# Введение в NLP. Обратная частота встречаемости TF-IDF

В работе используется движок полнотекстового поиска Postgres. Документы с помощью него индексируются по «лексемам» – базовым синтаксическим единицам представляющим неизменяемые части слов.

Результаты поиска ранжируются в соответствии с статистикой встречаемости слов во всей базе и в документе:

$$TF-IDF(w, d, C) = \frac{\text{count}(w, d)}{\text{count}(d)} * \log \frac{|C|}{\sum_{d \in C} \text{count}(d, w)}$$

Это можно назвать «важностью слова»

Пример разбиения в базе данных:

База знаний ответов на вопросы

Файл Справка

☐ Поиск и лексемизация только по вопросам ☒ Скрыть к...

База ответов Все Лексемы базы Релевантность и запросы Анализ запроса Результаты поиска

	Лексема Pg	Вес (Pg)	Вес	Ответов с вхождением	Вхождений за всю базу	
424	экзопланет	A	1.0	1	1	questions
425	экономик	A	1.0	1	1	questions
426	язык	A	1.0	1	1	questions
427	edu	B	0.4	5	9	abstract
428	pnu	B	0.4	5	9	abstract

При добавлении данных автоматически применяется операция «стемминг» к документу, и строится подобный индекс с подсчетом вхождения лексемы.

## **Обработка естественного языка**

Базовая машина полнотекстового поиска работает лучше стандартного поиска, однако не всегда достаточна для пользовательских запросов на естественном языке.

Чтобы повысить качество поиска нам необходимо углубиться в теорию работы с естественным языком (NLP – Natural Language Processing)

# Основы NLP анализа текста. Эмбединги

В широком смысле, **эмбединг** - это процесс преобразования каких-либо данных (чаще всего текста, но могут быть и изображения, звуки и т.д.) в набор чисел, **векторы**, которые машина может не только хранить, но и с которыми она может работать.

Именно преобразовав слово в числовой вид можно применить аппарат математики и вычислительной техники к NLP-анализу текста

Категория	тип	описание
Текстовые эмбединги	Word Embeddings	Эти эмбединги преобразуют слова в векторы, так что слова с похожим значением имеют похожие векторные представления
	Sentence Embeddings	Здесь уже идет дело о целых предложениях. Подобные модели создают векторные представления для целых предложений или даже абзацев, улавливая гораздо более тонкие нюансы языка.
Эмбединги изображений	CNN	CNN позволяет преобразовать изображения в векторы, которые затем используются для различных задач, например, классификации изображений или даже генерации новых изображений.
	Autoencoders	Автоэнкодеры могут сжимать изображения в более мелкие, плотные векторные представления, которые затем могут быть использованы для различных целей, включая декомпрессию или даже обнаружение аномалий.
Эмбединги для других типов данных	Graph Embeddings	Применяются для работы с графовыми структурами (к примеру рекомендательные системы). Это способ представить узлы и связи графа в виде векторов.
	Sequence Embeddings	Используются для анализа последовательностей, например, во временных рядах или в музыке.



**Векторные пространства** — это математические структуры, состоящие из векторов. Векторы можно понимать как точки в некотором пространстве, которые обладают направлением и величиной. В эмбедингах, каждый вектор представляет собой уникальное представление объекта, преобразованное в числовую форму.

**Размерность вектора** определяет, сколько координат используется для описания каждого вектора в пространстве. В эмбедингах высокая размерность может означать более детализированное представление данных. Векторное пространство для текстовых эмбедингов может иметь тысячи измерений.

**Расстояние между векторами** в эмбедингах измеряется с помощью метрик, таких как *Евклидово расстояние* или *косинусное сходство*. Метрики позволяют оценить, насколько близко или далеко друг от друга находятся различные объекты в векторном пространстве, что является основой для многих алгоритмов машинного обучения, таких как классификация

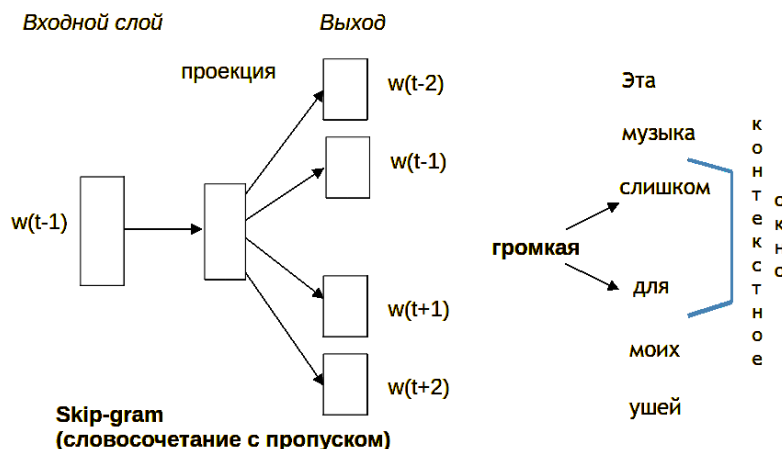
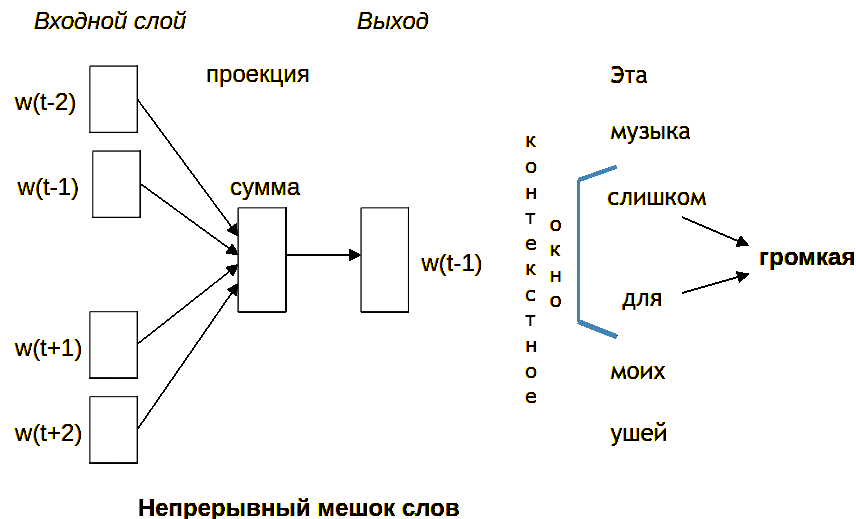
# Используемые технологии. Word2Vec

Word2Vec использует нейронные сети для обучения векторных представлений слов из больших наборов текстовых данных. Существуют две основные архитектуры Word2Vec:

**CBOW:** предсказывает текущее слово на основе контекста (окружающих слов). Например, в предложении "Собака лает на \_\_\_\_\_", CBOW попытается угадать недостающее слово (например, "почтальона") на основе окружающих слов.



**Skip-gram:** работает наоборот по сравнению с CBOW. Использует текущее слово для предсказания окружающих его слов в предложении. Например, если взять слово "кошка", модель попытается предсказать слова, которые часто встречаются в окружении слова "кошка", такие как "мышь", "мяукает" и т.д.



Ты не должен создавать машину по подобию разума человека

Скользящее окно перемещается по тексту

ты	не	должен	создавать	машину	по	подобию	разума	человека	...
ты	не	должен	создавать	машину	по	подобию	разума	человека	...
ты	не	должен	создавать	машину	по	подобию	разума	человека	...
ты	не	должен	создавать	машину	по	подобию	разума	человека	...
ты	не	должен	создавать	машину	по	подобию	разума	человека	...

Датасет

вход1	вход2	выход
не	должен	создавать
должен	создавать	машину
создавать	машину	по
машину	по	подобию
по	подобию	разума

предыдущий контекст

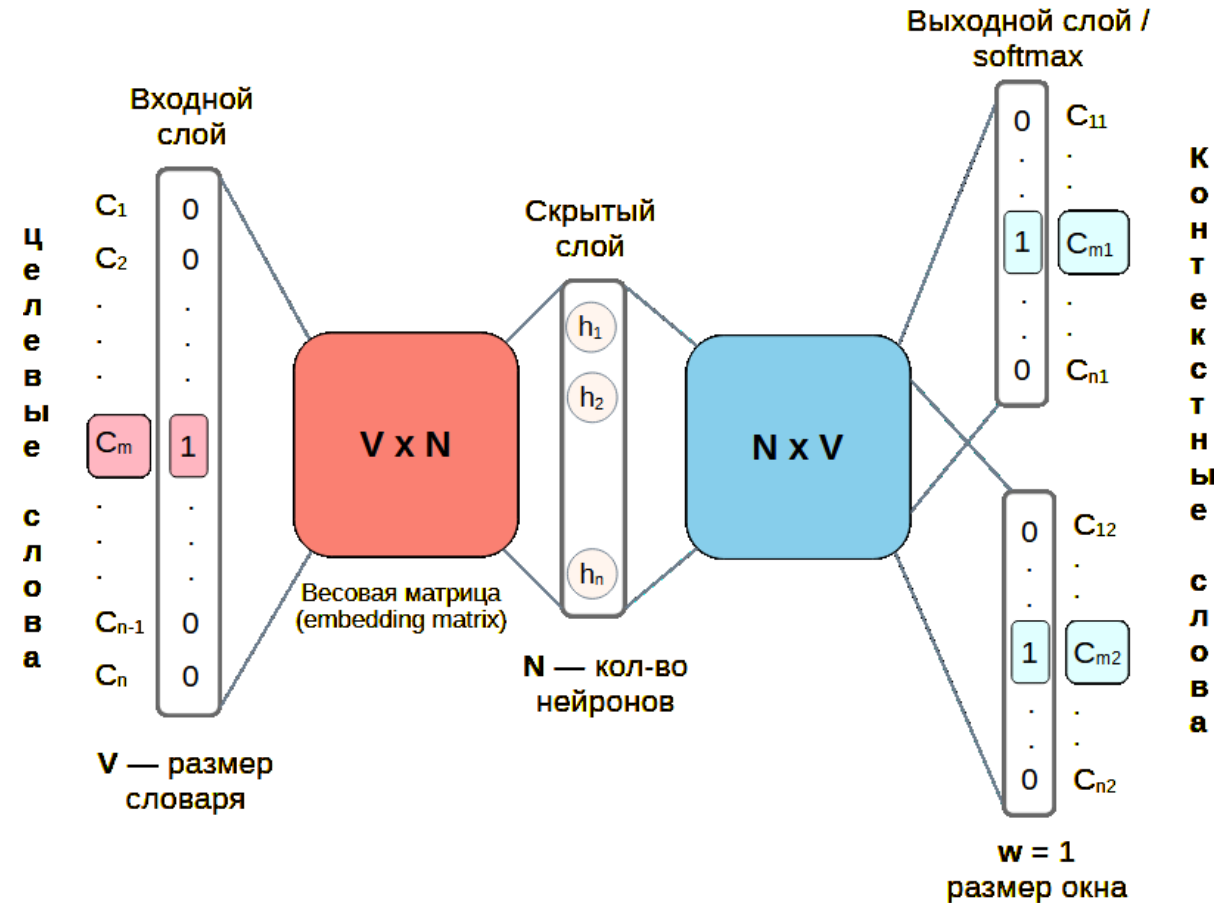
Архитектура модели	Набор тестов на семантико-синтаксическую взаимосвязь слов		Связанность слов MSR (тестовый набор[20])
	Семантическая точность, %	Синтаксическая точность, %	
RNNLM	9	36	35
NNLM	23	53	47
CBOW	24	64	61
Skip-gram	55	59	56

Ты не должен создавать машину по подобию разума человека

ты	не	должен	создавать	машину	по	подобию	разума	человека	...
----	----	--------	-----------	--------	----	---------	--------	----------	-----

входное слово	искомое слово
должен	ты
должен	не
должен	создавать
должен	машину

## Пример архитектуры Word2vec ИНС (skip-gram), 1 скрытый слой, окно = 1



$V$  - количество слов в словаре после обучения, каждое слово в словаре описывается как вектор с однократным кодированием (двоичный вектор, в котором только позиция соответствующего слова имеет значение 1),  $N$  - количество нейронов (размерность векторного пространства слов). Весовая матрица  $V \times N$  хранит обученный вектор и моделью предсказываются векторы которые соответствуют словам близким по контексту входному — то есть при обучении находившихся слева и справа в тексте (окно  $w=1$ ).

# Разработанная программа

База знаний ответов на вопросы

Файл Справка

☒ Поиск и лексемизация только по вопросам

☒ Скрыть консоль

База ответов

Все Лексемы базы

Релевантность и запросы

Анализ запроса

Результаты поиска

Исходный вопрос

Уголовный кодекс в Древней Руси?

Выполнить

Анализ ->

Добавить вопрос в список

Лемматизация и stemming от Postgres

уголовн кодекс древн рус

<- Выделить синонимы

Таблица исключений NER-токенов исходного запроса (NamedEntities)

Выделенные леммы в запросе на которые присутствуют синонимы в базе

Уголовный<уголовн> [tf\_idf: 0.00][часть речи: ADJ(прил.)]:  
-внести<внести>, [tf\_idf: 0.14][сходство: 0.64]

кодeкс<кодeкс> [tf\_idf: 0.00][часть речи: NOUN(сущ.)]:  
-закон<закон>, [tf\_idf: 0.68][сходство: 0.76]  
-документ<документ>, [tf\_idf: 0.10][сходство: 0.69]

Древней<древн> [tf\_idf: 0.68][часть речи: ADJ(прил.)]:

Таблица исключений словосочетаний исходного запроса (Bigram)

Bigram

Начало

1 уголовный кодекс 0

2 древней руси 19

VVVV---Оптимизировать---VVVV

Рекомендованный поисковой запрос (оптимизированный для базы знаний)

закон древн рус

Выполнить

Очистить всё

Результат анализа запроса

Синтаксический анализ

Члены предложения:

1-е предложение

Основные члены предложения:

- кодeкс - сказуемое

Второстепенные члены предложения:

- Уголовный - определение
- Древней Руси - определение
- Древней Руси - дополнение

Сокращение предложения, до 3 уровня связи членов предлоа

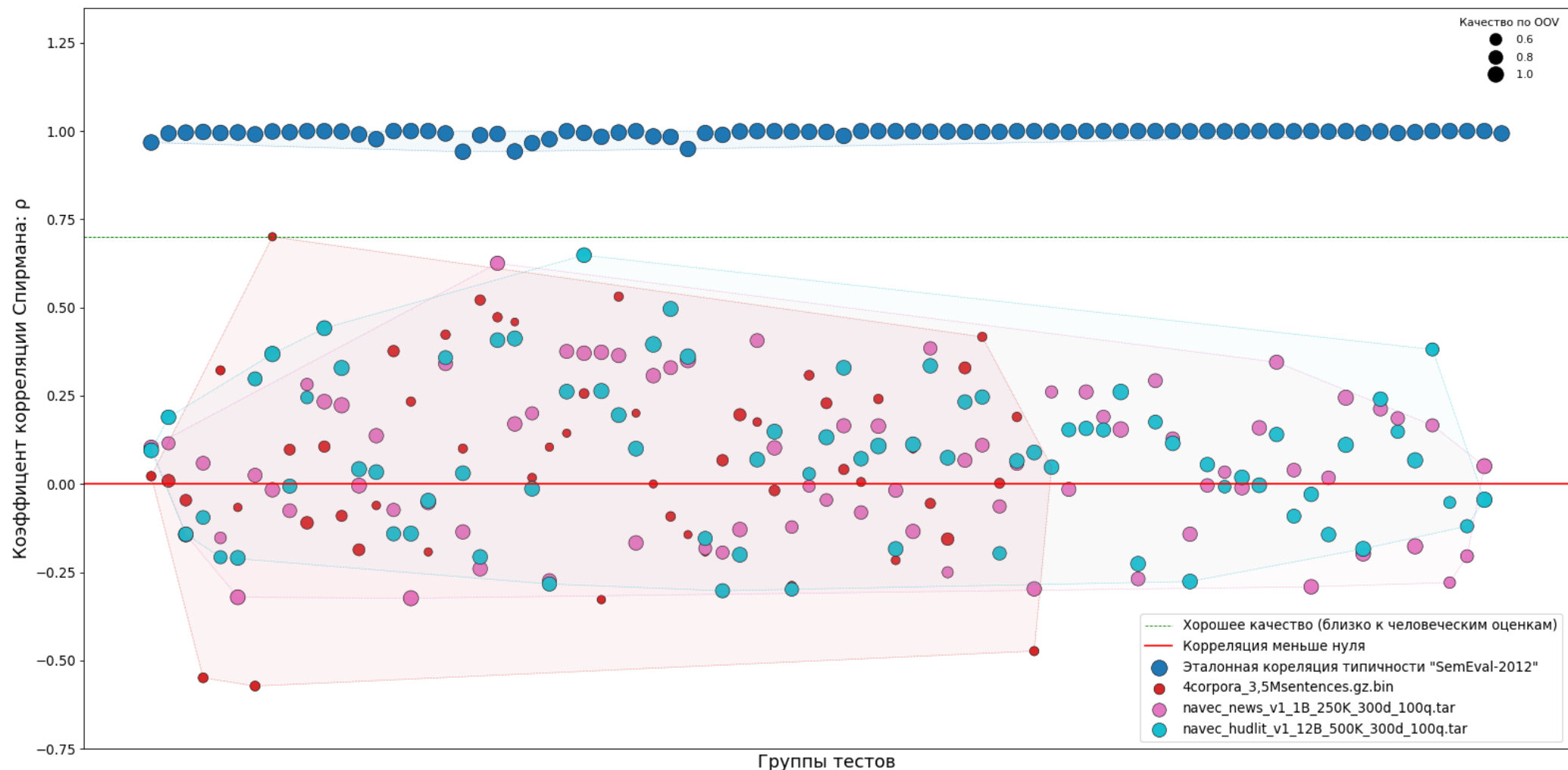
Уголовный кодекс Древней Руси

Программа использует комплекс из двух алгоритмов: «Алгоритм синтаксического анализа запроса, выявление основной части запроса» и «Алгоритм оптимизации по семантической близости и TF-IDF» для модификации пользовательского запроса к базе данных.

В работе алгоритма используется как уже обученные модели из пакета gensim и natasha, так и полностью самостоятельно обученная фразовая Word2Vec модель словосочетаний

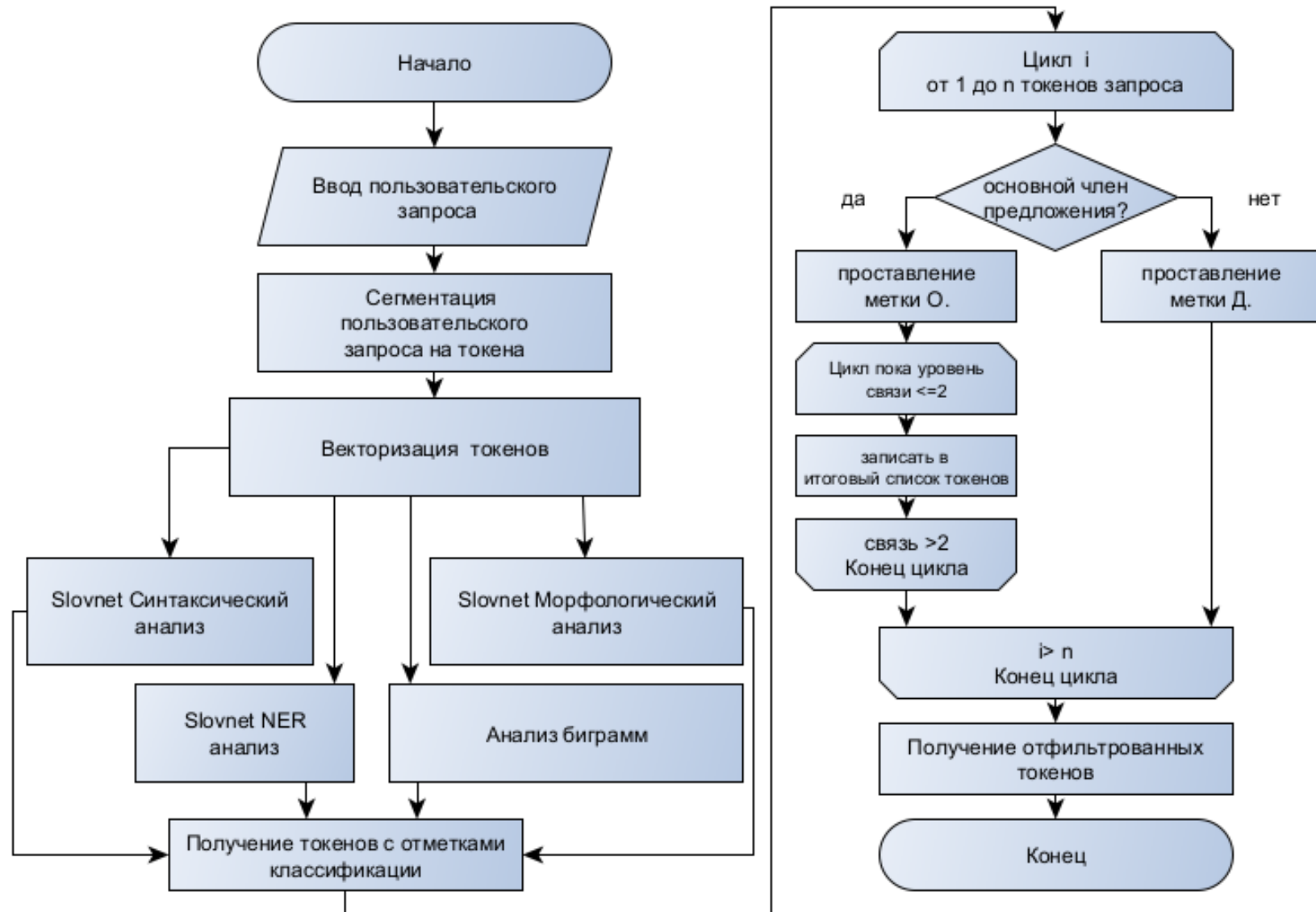
# Семантическая модель языка используемая в программе

Тесты семантической близости "SemEval-2012-Platinum-Ratings"  
(3 модели)

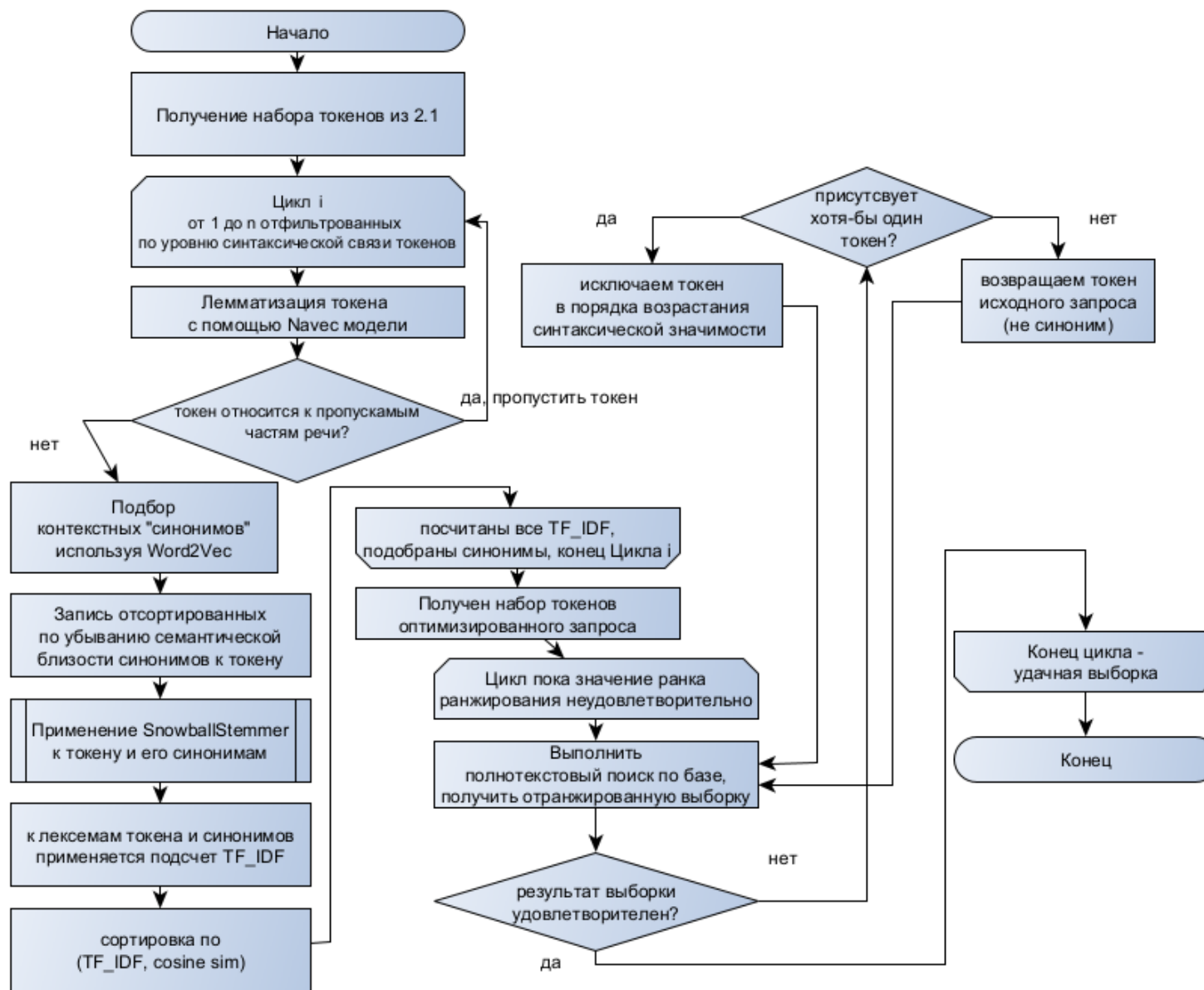


Как видно из пузырьковой диаграммы, лучше всего улавливает семантическую близость модель naves обученная на корпусах из 12 миллиардов слов художественной литературы. Обученная мной модель 4corpora\_3,5Msentences иногда и давала результат порядка 0.75, что считается хорошим качеством, однако в данном наборе слов модели удалось найти векторное представление пар слов в лучшем случае в 50% случаев.

# «Алгоритм синтаксического анализа запроса, выявление основной части запроса»



## «Алгоритм оптимизации по семантической близости и TF-IDF»





## Полученные результаты эффективности

Ранжирование — задача сортировки набора элементов из соображения их релевантности. Чаще всего релевантность понимается по отношению к некому объекту. В задаче информационного поиска объект — это запрос, элементы — всевозможные документы (ссылки на них), а релевантность — соответствие документа запросу.

Для релевантности существует метрика: Средняя точность на k-элементах (map@K)

Были проведены расчеты для 10 поисковых запросов с размеченной релевантностью на базе из 100 вопросов:

Тип поиска	map@K
Поиск по вхождению строки	0,2
Полнотекстовый поиск Postgres	0,4
Полнотекстовый поиск с NLP оптимизацией	0,861



# Вопросы