Министерство образования Республики Беларусь

Учреждение образования “Белорусский государственный университет информатики и радиоэлектроники”

Факультет информационных технологий и управления

Кафедра интеллектуальных информационных технологий

Отчёт по дисциплине

“Естественно-языковой интерфейс интеллектуальных систем”

Лабораторная работа №5

“Естественно-языковой интерфейс диалоговой системы с пользователем”

Выполнили студенты группы 221702: Целуйко Д.А.

Гринь Н.А.

Проверил: Крапивин Ю.Б.

Минск 2025

Лабораторная работа №5

**Цель работы:**

Освоить принципы проектирования и разработки

естественно-языковых интерфейсов с пользователем в диалоговых системах.

**Задачи лабораторной работы (Вариант 13):**

1. Изучить основы моделирования процесса общения.

2. Приобрести навыки проектирования элементов диалоговых систем.

**Язык текста:**

Английский

**Предметная область:**

Кинематограф

**Используемые средства разработки:**

Для решения поставленной задачи было разработано десктоп-приложение на языке программирования **Python** с использованием библиотеки **Tkinter**, а также некоторые другие библиотеки

**Tkinter —** это пакет для Python, предназначенный для работы с библиотекой Tk. Библиотека Tk содержит компоненты графического интерфейса пользователя (graphical user interface – GUI). Эта библиотека написана на языке программирования Tcl.

**Spacy (spaCy)** — это программная библиотека с открытым исходным кодом для расширенной обработки естественного языка, написанная на языках программирования Python и Cython. Были догружены следующие ресурсы:

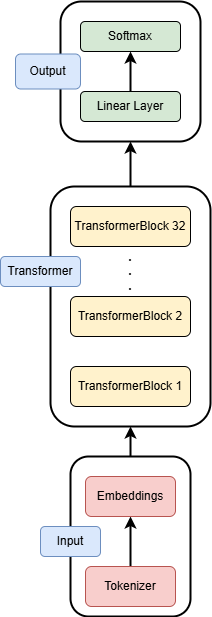
* **spacy.load('en\_core\_web\_sm') -** легкая английская языковая модель, оптимизированная для использования центральным процессором. Эта модель позволяет выполнять присвоение частей речи, синтаксический анализ зависимостей и лемматизацию токенов во входном тексте. Разработано компанией Explosion AI.

**NLTK (Natural Language Toolkit) -** пакет библиотек и программ для символьной и статистической обработки естественного языка, написанных на языке программирования Python.

**Ollama** — это инструмент (фреймворк) с открытым исходным кодом, который значительно упрощает процесс скачивания, настройки и запуска больших языковых моделей, таких как Llama 3, Mistral, Mixtral, Phi и многих других, локально на вашем собственном компьютере (Windows, macOS, Linux). Установка для Linux до банального проста, необходимо выполнить в терминале команду **curl -fsSL https://ollama.com/install.sh | sh**.

Для Windows и MacOS существуют приложения, которые можно скачать с [официального сайта](https://ollama.com/download) выбрав свою ОС.

**Llama 3** (Large Language Model Meta AI 3) — это семейство больших языковых моделей (LLM) с открытым исходным кодом (open source), разработанное и выпущенное компанией Meta AI (подразделение компании Meta, ранее Facebook) в апреле 2024 года. Это следующее поколение моделей после Llama 2. Установка через ollama заключается в выполнении в терминале команды ollama pull “имя\_модели” (без ковычек), в данном случае llama3. Алгоритм работы llama3:

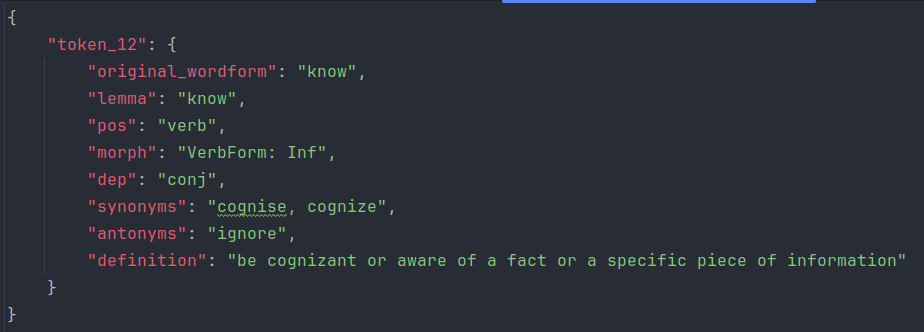


Более подробно можно почитать в [статье на хабре](https://habr.com/ru/articles/879468/).

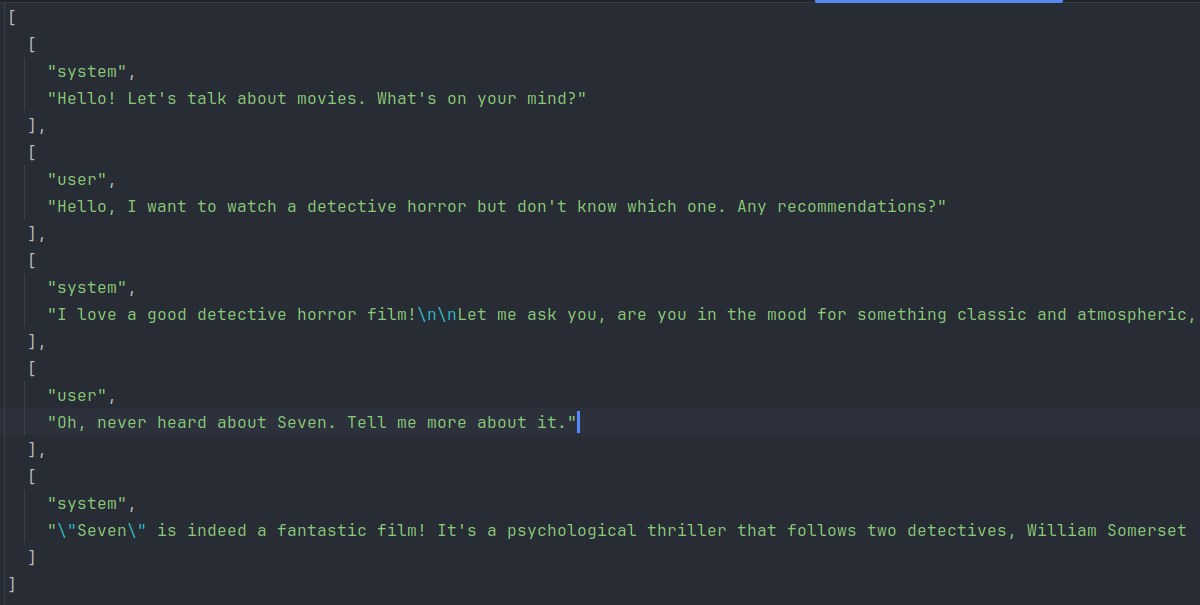
**Были догружены следующие ресурсы:**

* **wordnet -** это основной компонент, используемый для получения определений, синонимов и антонимов, которые требовалось добавить в новые столбцы таблицы.
* **punkt** - Хотя конкретно это приложение полагается на spaCy для большей части токенизации и разделения предложений, многие функции NLTK (включая некоторые потенциальные взаимодействия с WordNet или анализ его определений) зависят от punkt. Его загрузка часто необходима для корректной работы NLTK, даже если он не вызывается напрямую в нашем текущем коде.
* **averaged\_percepetron\_tagger** **-** Поиск в WordNet сильно зависит от части речи слова (например, "bank" как существительное и "bank" как глагол — это разные значения). Хотя мы в основном используем POS-теги spaCy и преобразуем их в теги WordNet (\_map\_spacy\_pos\_to\_wordnet), наличие собственного теггера NLTK гарантирует, что NLTK сможет выполнить разметку, если это потребуется для внутренних операций. Это стандартная практика при использовании интерфейса WordNet в NLTK.
* **omw-1.4 -** Даже при использовании в основном английского WordNet через NLTK, omw-1.4 часто предоставляет обновленные соответствия и обеспечивает лучшую совместимость и доступ к связанным структурам данных внутри модуля wordnet NLTK. Обычно рекомендуется загружать его вместе с wordnet для достижения наилучших результатов.

**Json** — используется в качестве стандартного текстового формата для импорта/экспорта информации об анализируемых словах и их морфологических и синтаксических данных.



*Рис. 1 – Пример хранения токенов в формате JSON*

**

*Рис. 2 – Пример хранения истории чата в формате JSON*

Система работает со следующими частями речи:

1. **VERB**: глагол
2. **PRON**: местоимение
3. **ADJ**: прилагательное
4. **NOUN**: существительное
5. **ADV**: наречие
6. **ADP**: предлог
7. **DET**: артикль/детерминатив
8. **PROPN**: имя собственное
9. **CCONJ**: сочинительный союз
10. **AUX**: вспомогательный глагол
11. **SCONJ**: подчинительный союз
12. **NUM**: числительное
13. **PART**: частица
14. **INTJ**: междометие

Система учитывает следующие синтаксические отношения между членами предложения:  
1**. ROOT (Корень)**: Основной элемент предложения, от которого зависят другие слова.

* + Пример: "The quick brown fox **jumps**." (Здесь "jumps" — корень.)

1. **nsubj (Подлежащее)**: Основное подлежащее предложения.
   * Пример: "**The cat** is sleeping." (Здесь "The cat" — подлежащее.)
2. **dobj (Прямое дополнение)**: Объект, на который непосредственно воздействует глагол.
   * Пример: "She ate **the apple**." (Здесь "the apple" — прямое дополнение.)
3. **iobj (Косвенное дополнение)**: Объект, который получает действие, но не является прямым объектом.
   * Пример: "He gave **her** a book." (Здесь "her" — косвенное дополнение.)
4. **pobj (Объект предлога)**: Объект, управляемый предлогом.
   * Пример: "She sat **on the chair**." (Здесь "the chair" — объект предлога "on".)
5. **subj (Именная часть)**: Подлежащее в именной части предложения.
   * Пример: "**The book** on the table is interesting." (Здесь "The book" — именная часть.)
6. **csubj (Подлежащее придаточного предложения)**: Подлежащее в придаточном предложении.
   * Пример: "I think **she is smart**." (Здесь "she" — подлежащее придаточного предложения.)
7. **attr (Атрибут)**: Атрибут или определение, относящийся к подлежащему.
   * Пример: "The book is **interesting**." (Здесь "interesting" — атрибут.)
8. **agent (Агент)**: Выполняет действие в пассивной конструкции.
   * Пример: "The cake was eaten **by her**." (Здесь "her" — агент.)
9. **advcl (Наречное дополнение)**: Наречное дополнение, модифицирующее глагол.
   * Пример: "He left **because he was tired**." (Здесь "because he was tired" — наречное дополнение.)
10. **advmod (Наречие)**: Наречие, модифицирующее глагол, прилагательное или другое наречие.
    * Пример: "She runs **quickly**." (Здесь "quickly" — наречие.)
11. **amod (Определение)**: Прилагательное, модифицирующее существительное.
    * Пример: "The **big** dog." (Здесь "big" — определение.)
12. **appos (Приложение)**: Приложение, добавляющее информацию к существительному.
    * Пример: "My friend, **a doctor**, lives nearby." (Здесь "a doctor" — приложение.)
13. **aux (Вспомогательный глагол)**: Вспомогательный глагол, помогающий основному глаголу.
    * Пример: "She **is** eating." (Здесь "is" — вспомогательный глагол.)
14. **auxpass (Вспомогательный глагол в пассиве)**: Вспомогательный глагол в пассивной конструкции.
    * Пример: "The cake **was** eaten." (Здесь "was" — вспомогательный глагол в пассиве.)
15. **cc (Сочинительный союз)**: Союз, соединяющий элементы на одном уровне.
    * Пример: "apples **and** oranges." (Здесь "and" — сочинительный союз.)
16. **ccomp (Дополнение придаточного предложения)**: Придаточное предложение, выполняющее роль дополнения.
    * Пример: "I know **that she is smart**." (Здесь "that she is smart" — дополнение придаточного предложения.)
17. **compound (Сложное слово)**: Часть составного слова.
    * Пример: "coffee **cup**." (Здесь "cup" — часть сложного слова.)
18. **conj (Соединительный элемент)**: Элемент, соединенный с другим элементом.
    * Пример: "I like apples **and** oranges." (Здесь "oranges" — соединительный элемент.)
19. **dative (Дательный падеж)**: Косвенное дополнение, получающее действие.
    * Пример: "She gave **him** a book." (Здесь "him" — дательный падеж.)
20. **dep (Неопределенная зависимость)**: Неопределенная зависимость, не подходящая под другие категории.
    * Пример: Может использоваться в сложных конструкциях, где зависимость не ясна.
21. **det (Определитель)**: Определитель, уточняющий существительное.
    * Пример: "**The** book." (Здесь "The" — определитель.)
22. **expl (Вводное слово)**: Вводное слово, не несущее смысловой нагрузки.
    * Пример: "**There** is a book." (Здесь "There" — вводное слово.)
23. **intj (Междометие)**: Междометие, выражающее эмоцию.
    * Пример: "**Oh**, I see." (Здесь "Oh" — междометие.)
24. **mark (Маркер)**: Маркер, вводящий придаточное предложение.
    * Пример: "I think **that** she is smart." (Здесь "that" — маркер.)
25. **meta (Мета-модификатор)**: Модификатор, изменяющий структуру предложения.
    * Пример: "She is **not** coming." (Здесь "not" — мета-модификатор.)
26. **neg (Отрицание)**: Отрицание, изменяющее смысл предложения.
    * Пример: "She is **not** coming." (Здесь "not" — отрицание.)
27. **nmod (Именное дополнение)**: Дополнение, модифицирующее существительное.
    * Пример: "The book **on the table**." (Здесь "on the table" — именное дополнение.)
28. **npadvmod (Именная группа как наречие)**: Именная группа, выполняющая роль наречия.
    * Пример: "She left **this morning**." (Здесь "this morning" — именная группа как наречие.)
29. **nummod (Числительное)**: Числительное, модифицирующее существительное.
    * Пример: "**Three** books." (Здесь "Three" — числительное.)
30. **oprd (Объектное дополнение)**: Объектное дополнение, связанное с подлежащим.
    * Пример: "She considers him **a friend**." (Здесь "a friend" — объектное дополнение.)
31. **parataxis (Паратаксис)**: Паратаксис, соединяющий предложения без явного союза.
    * Пример: "He came; **she left**." (Здесь "she left" — паратаксис.)
32. **pcomp (Дополнение предлога)**: Дополнение, управляемое предлогом.
    * Пример: "She is afraid **of spiders**." (Здесь "of spiders" — дополнение предлога.)
33. **poss (Притяжательное дополнение)**: Притяжательное дополнение, указывающее на владение.
    * Пример: "**Her** book." (Здесь "Her" — притяжательное дополнение.)
34. **preconj (Предшествующий союз)**: Союз, предшествующий соединенному элементу.
    * Пример: "She is **both** smart **and** kind." (Здесь "both" — предшествующий союз.)
35. **predet (Пред-определитель)**: Определитель, предшествующий основному определителю.
    * Пример: "**All the** books." (Здесь "All" — пред-определитель.)
36. **prep (Предлог)**: Предлог, управляющий объектом.
    * Пример: "She sat **on** the chair." (Здесь "on" — предлог.)
37. **prt (Частица)**: Частица, изменяющая значение глагола.
    * Пример: "She gave **up**." (Здесь "up" — частица.)
38. **punct (Пунктуация)**: Пунктуация, структурирующая предложение.
    * Пример: "Hello, world!" (Здесь "," и "!" — пунктуация.)
39. **quantmod (Квантификатор)**: Квантификатор, указывающий на количество.
    * Пример: "**Many** books." (Здесь "Many" — квантификатор.)
40. **relcl (Относительное дополнение)**: Придаточное предложение, модифицирующее существительное.
    * Пример: "The book **that she read**." (Здесь "that she read" — относительное дополнение.)
41. **xcomp (Открытое дополнение придаточного предложения)**: Придаточное предложение, выполняющее роль открытого дополнения.
    * Пример: "She started **to read**." (Здесь "to read" — открытое дополнение придаточного предложения.)

**Планируемые для реализации алгоритмы:**

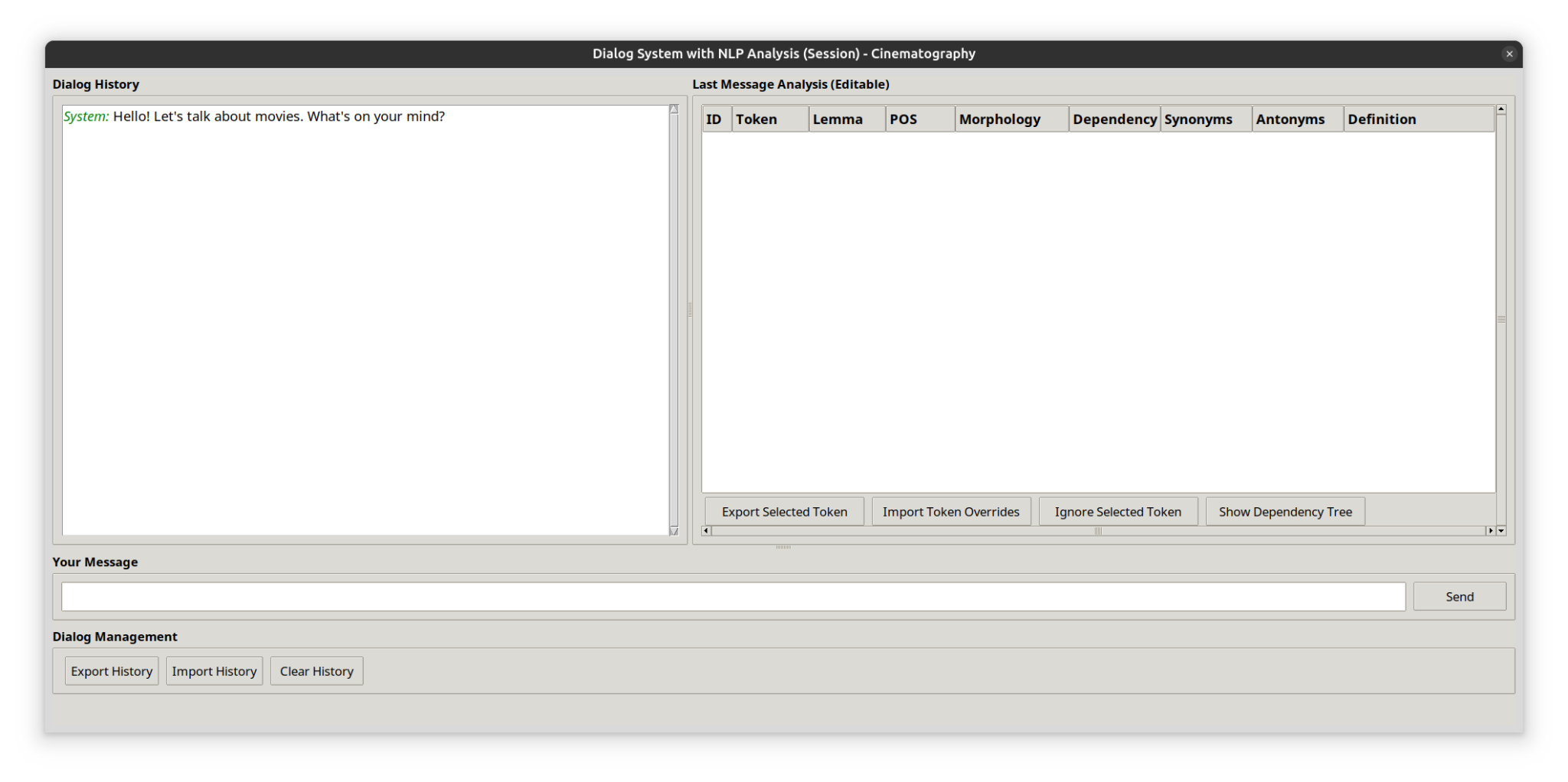
* Алгоритм Морфологического анализа
* Алгоритм "Синтаксический анализ и отображение"
* Алгоритм "Семантический анализ и отображение (на основе WordNet)"

**Что из себя представляет система?**Разработанное приложение представляет собой сеансовую диалоговую систему с графическим интерфейсом пользователя (GUI), реализованную на Python с использованием библиотеки Tkinter. Система предназначена для взаимодействия с пользователем на английском языке в предметной области "Кинематограф".

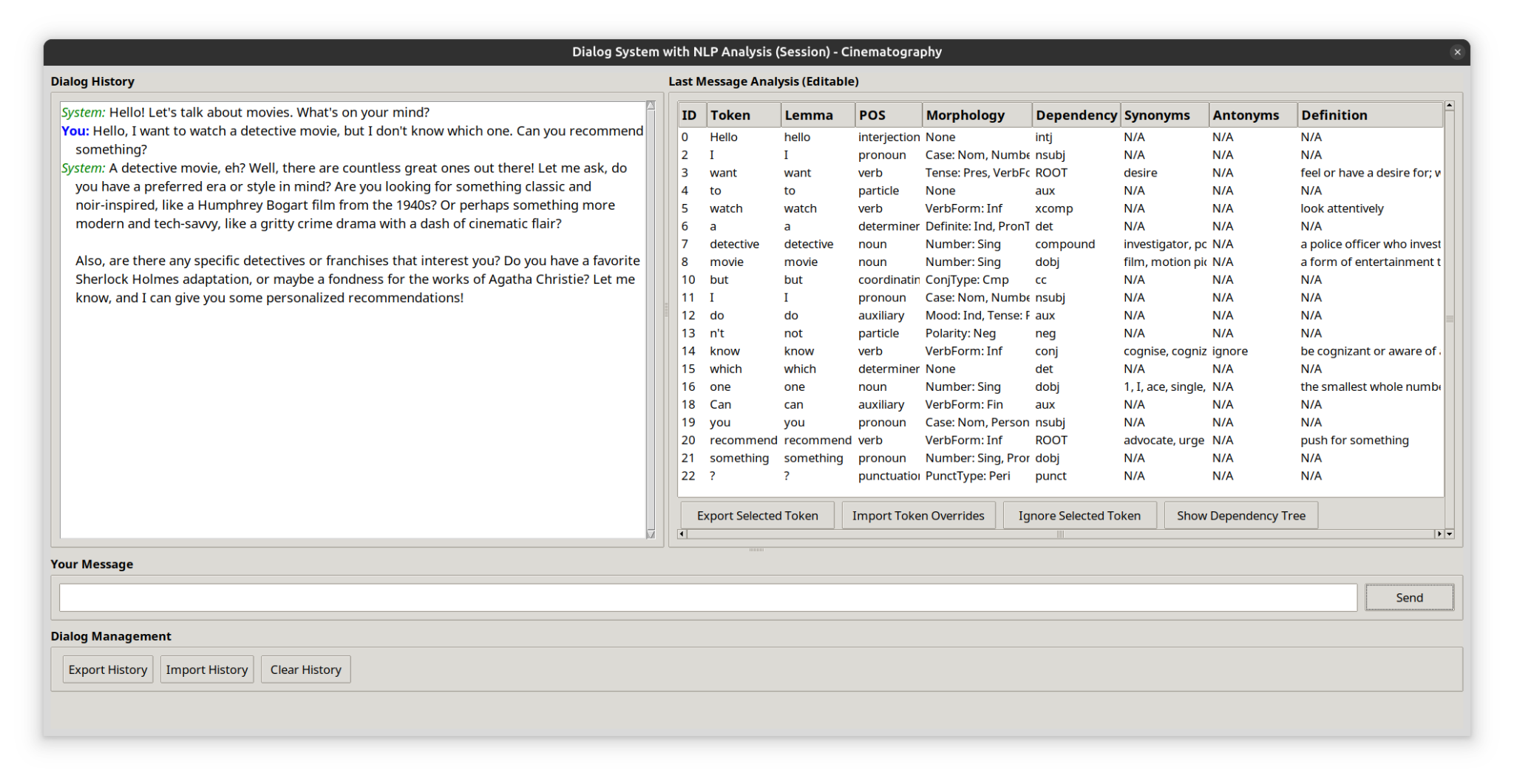
**Ключевые функции:**

* **Диалоговое взаимодействие:** Пользователь вводит текстовые сообщения, система генерирует и отображает ответные реплики, поддерживая простую беседу о кино. История диалога отображается в специальном окне.
* **NLP-анализ:** Последнее сообщение пользователя автоматически анализируется с помощью библиотеки spaCy для получения детальной лингвистической информации:
  + Токенизация (разбиение на слова/знаки).
  + Лемматизация (приведение к базовой форме).
  + Части речи (POS-тегирование).
  + Морфологические признаки.
  + Синтаксические зависимости (Dependency parsing).
* **Семантический анализ:** Система извлекает синонимы, антонимы и определения для слов из сообщения пользователя.
* **Визуализация анализа:** Результаты морфологического, синтаксического и семантического анализа отображаются в интерактивной таблице. Также возможно построение визуального дерева синтаксических зависимостей для выбранного предложения.
* **Редактирование (сеансовое):** Пользователь может редактировать (переопределять) основные результаты анализа (лемма, POS, морфология, зависимость) для любого токена в таблице. Эти изменения действительны только для текущей сессии.
* **Импорт/Экспорт:** Реализована функция экспорта детального анализа выбранного токена (включая ручные правки) в файл JSON, а также импорта таких правок из JSON для их применения в текущей сессии. Также реализован импорт/экспорт истории чата.
* **Сеансовая память:** Вся история диалога и результаты анализа хранятся только во время работы приложения и не сохраняются между запусками (кроме явно экспортированных данных).
* **Помощь пользователю:** Элементы интерфейса снабжены всплывающими подсказками (Hovertips).

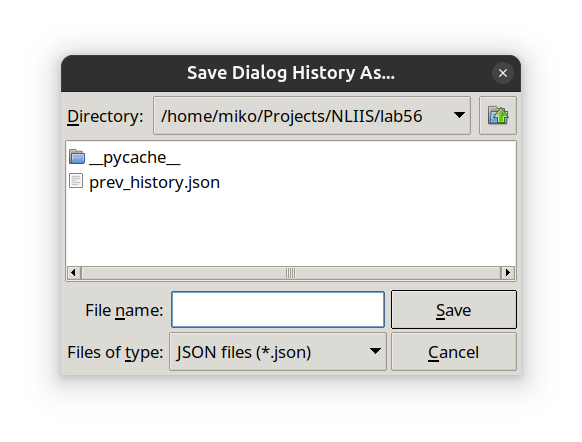
**Скриншоты разработанной системы:**



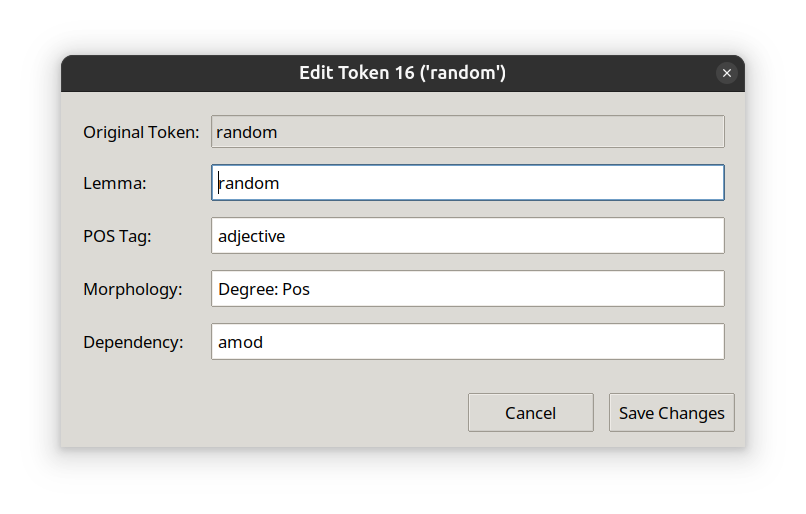
*Рис. 1- начальное окно приложения*

****

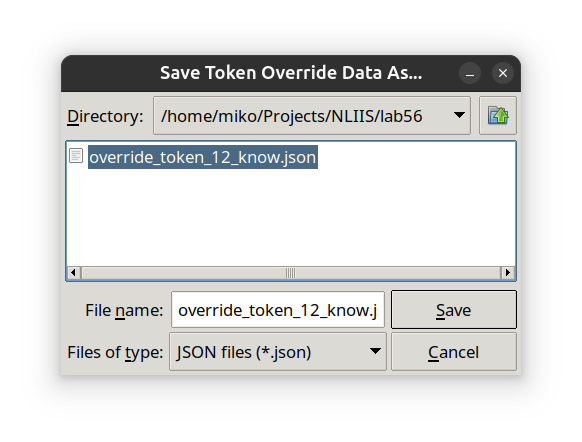
*Рис. 2 – ответ на вопрос пользователя*

**

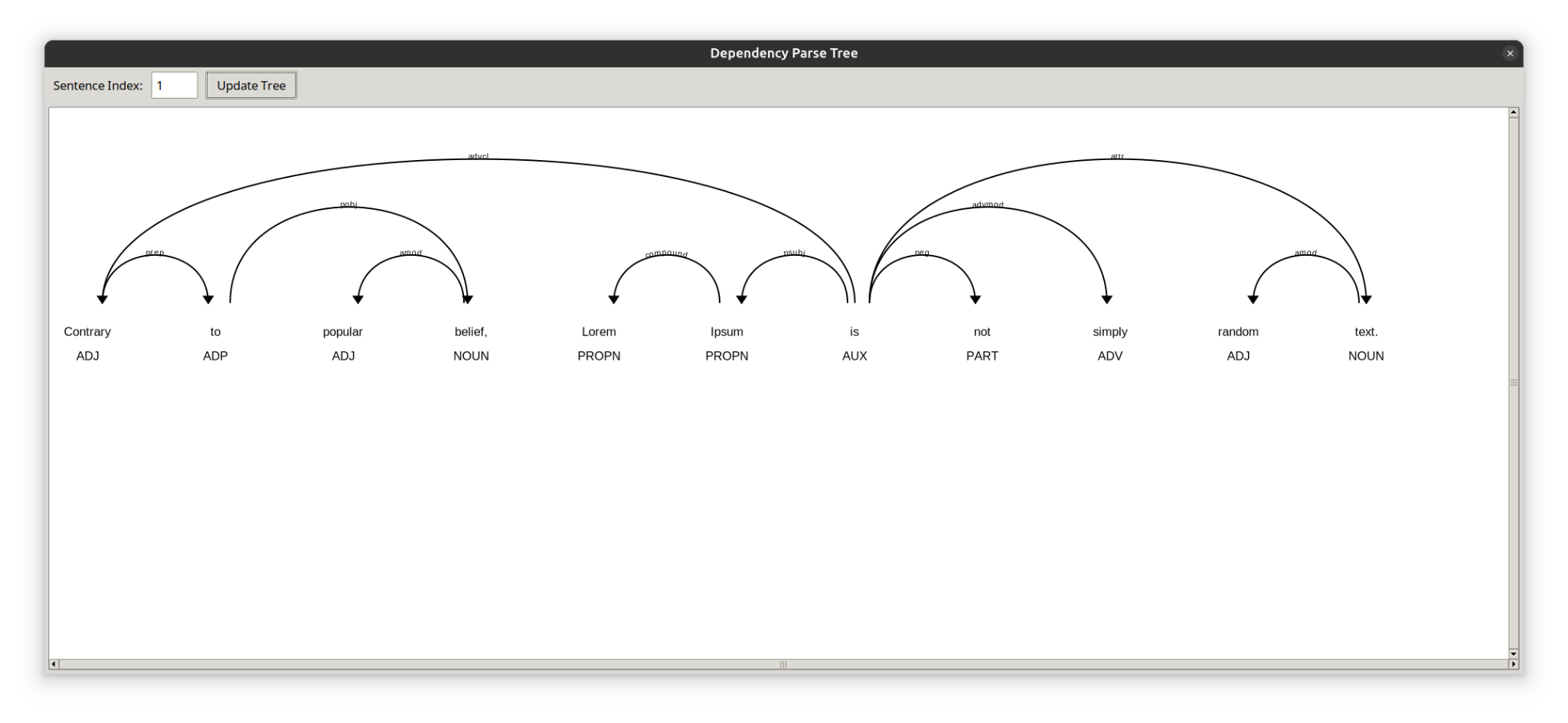
*Рис. 3 – Окно экспорта истории чата*

****

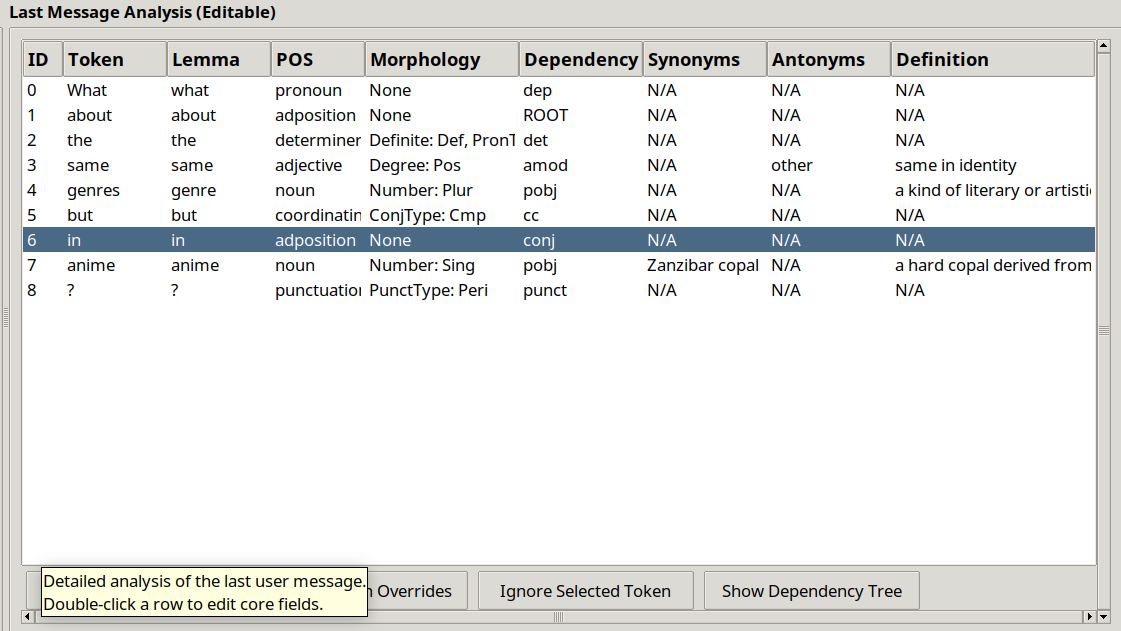
*Рис. 4 – Окно редактирования слова*

****

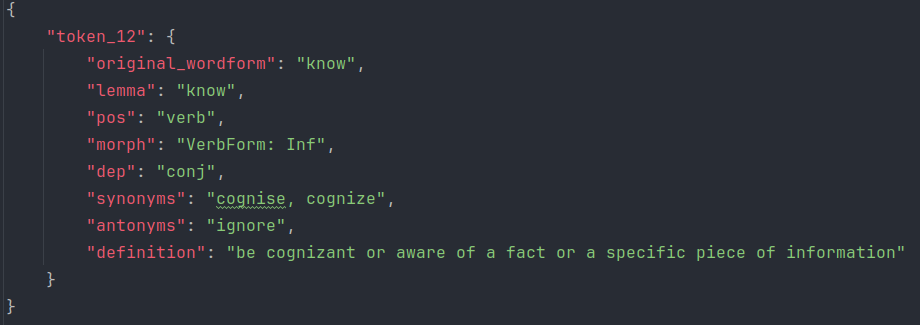
*Рис. 5 – Окно экспорта на примере слова “know” в json файл*



*Рис. 6 – синтаксическое дерево предложения “The quick brown fox jumps over the lazy dog.”*

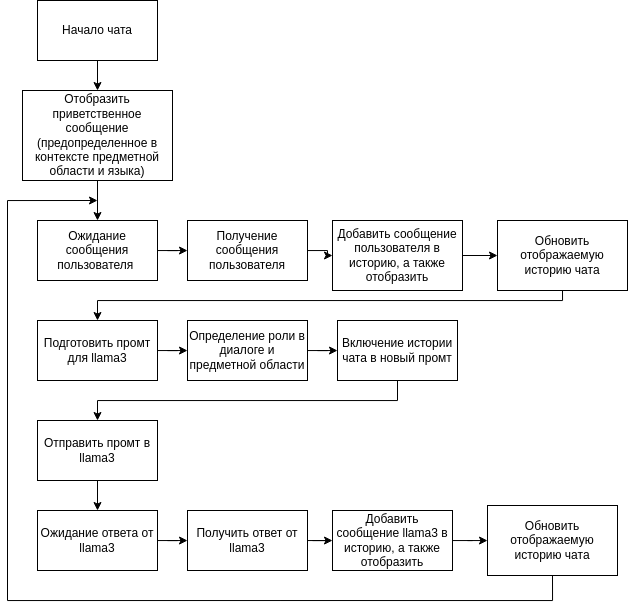
****

*Рис. 7 – пример окна с помощью к таблице*

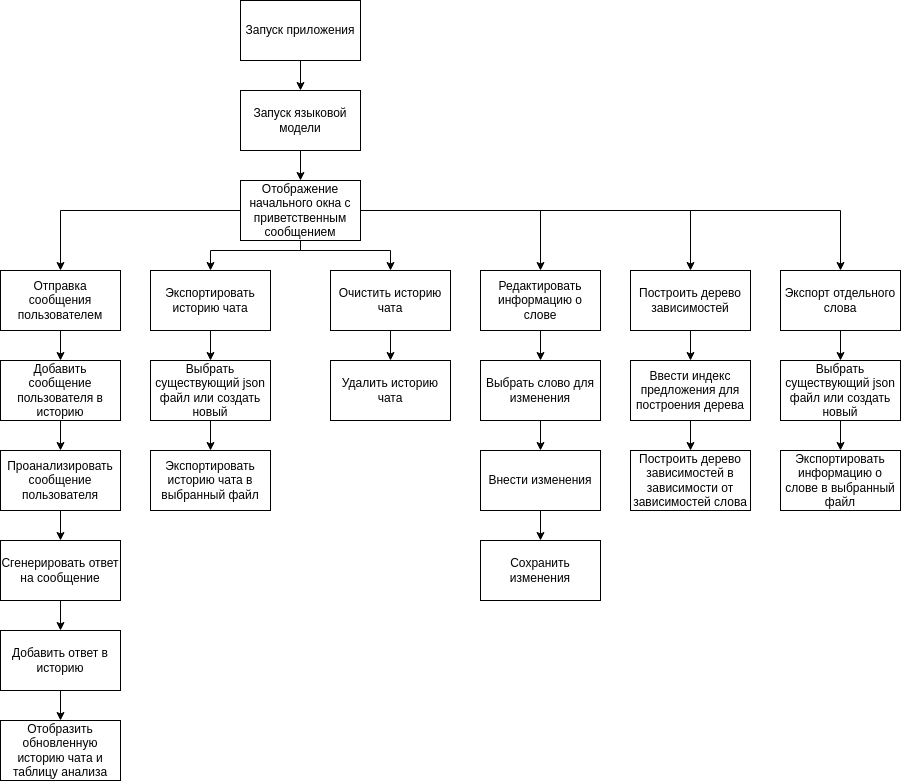
**

*Рис. 8 – Содержимое файла override\_token\_12\_know.json с экспортированным словом know*

**Схема алгоритма ведения диалога с пользователем:**

****

**Структурно-функциональная схема разрабатываемого приложения:**



*Рис. 9 - структурно-функциональная схема приложения*

**Выводы по работе и по перспективам развития приложения:**

Разработанное приложение представляет собой сеансовую диалоговую систему с графическим интерфейсом пользователя (GUI), реализованную на Python с использованием библиотеки Tkinter. Система предназначена для взаимодействия с пользователем на английском языке в предметной области "Кинематограф". В рамках перспективы развития приложения можно выделить возможность развернуть приложение на сервере с хорошей производительностью, а также использование модели с большим количеством параметров (использование сложной модели на персональном устройстве является практически невозможным из-за очень высоких требований к характеристикам железа).