Задача: подобрать к заданному слову ранжированный список гиперонимов.

Можно было использовать несколько вариантов обучения модели, однако, было решено использовать классификацию текстов. Для реализации данного метода было решено рассмотреть несколько вариантов нейронных сетей от самых простых, до более сложных: CNN, CNN + LSTM, BERT.

Для обучения модели необходимы положительные и отрицательные примеры. В качестве начальных данных были даны списки слов из различных источников: ruWordNet, Russian Wikipedia, библиотеки “Либрусек” и т.д. Но они представлены в разных видах и хранят много избыточной информации, например, id слов, другие типы связей между словами, например, синонимы, антонимы и т.п.

Модель для обучения можно было сформировать несколькими способами. Первый вариант заключается в том, чтобы конкатенировать пары гипоним - гипероним в строку. Такой формат данных отлично подходит для обучения CNN и CNN + LSTM, однако, не подходит для обучения Bert, потому как он “смотрит” на текст целиком, а не на отдельные слова. Второй способ представления слов - с контекстом. было решено найти заданные слова в тексте русской Wikipedia и использовать предложения из данного источника как текст.

Также для обучения модели необходимы отрицательные примеры. Одним из вариантов было взять случайное слово из базы. Однако, при данном методе вероятно, что мы попадем на такой же гипероним, так как гипероним гиперонима является гиперонимом. Для решения данной проблемы было решено взять вместо случайного слова слово-антоним. Таким образом мы уменьшаем вероятность того, что возьмем гипероним.

Задача состоит в нахождении подходящих гипероним из предложенного списка, но мы не можем при поиске слова-гиперонима составлять пары ( искать контекст) со всеми имеющимися в словаре словами (при желании желании, конечно, можем, Но это нерационально и занимает слишком много времени). Поэтому было предложено выполнять предобработку данных, т.е. сначала мы быстрым алгоритмом, основанном на вычислении векторов (fasttech) находим 200 слов, близких к заданному. При этом принимается гипотеза, что гиперонимы и гипонимы лежат достаточно близко друг к другу. После такой предобработки, уже применяется более тяжёлый алгоритм: один из предложенных методов.

Однако, в ходе работы было выявлено, что гипотеза “ближайших векторов” ошибочна. Потому что ближайшие вектора находятся не по смыслу, а по написанию.

Обучение CNN происходит порядка 705 секунд (на 7 итерациях). На вход при этом подается 7915 примеров, из которых 6332 идут на обучение модели и 1583 на ее валидацию (20 %).