Методы предобработки и векторизации текстов. Языковые модели.

Предобработка текста

- Для построения моделей машинного обучения необходимо предварительно предобработать текст. На этапе предобработки, как правило, используется только лингвистическая информация. Предобработка может включать следующие шаги:
  1. Уровень символов:
     - Токенизация – разбиение текста на токены (как правило, предложения или слова).
  2. Уровень слов:
     - Нормализация текста – лемматизация или стемминг .
     - Если текст получен путем генерации из аудио, то возможно к словам придется применять фонетические алгоритмы и искать наиболее похожие в словаре слова с использованием расстояния Левенштейна.
     - Частеречная разметка (Part-Of-Speech tagging, POS-tagging).
     - Выделение (распознавание) именованных сущностей, named-entity recognition (NER).
  3. Уровень предложений:
     - Разбор предложения, выделение семантических ролей.
- *Пример предобработки текста.*

Векторные представления на основе модели мешка слов

- Модель мешка слов.
- Использование CountVectorizer для построения мешка слов.
- Метрика TF-IDF.
- Использование TfidfVectorizer для построения мешка слов.
- *Пример решения задачи анализа тональности текста с помощью модели мешка слов.*

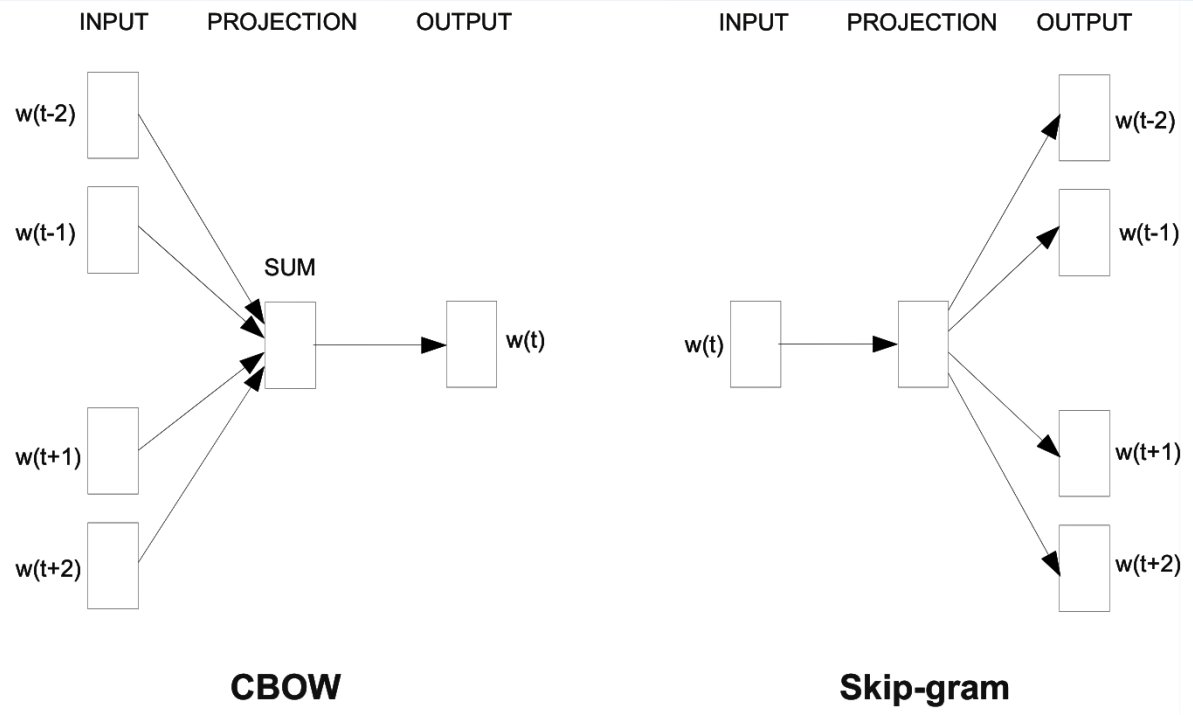
Неглубокие семантические векторные представления слов и документов

- *Для рассматриваемых подходов используется термин «неглубокие векторные представления», так как в настоящее время существуют более сложные модели векторизации (например, BERT), основанные на глубоком обучении.*
- Обзор моделей
- Семантические векторные представления слов:
  - Модель word2vec
  - Модель GloVe
  - Модель fastText
- Семантические векторные представления документов:
  - Модель doc2vec

Модель word2vec – описание

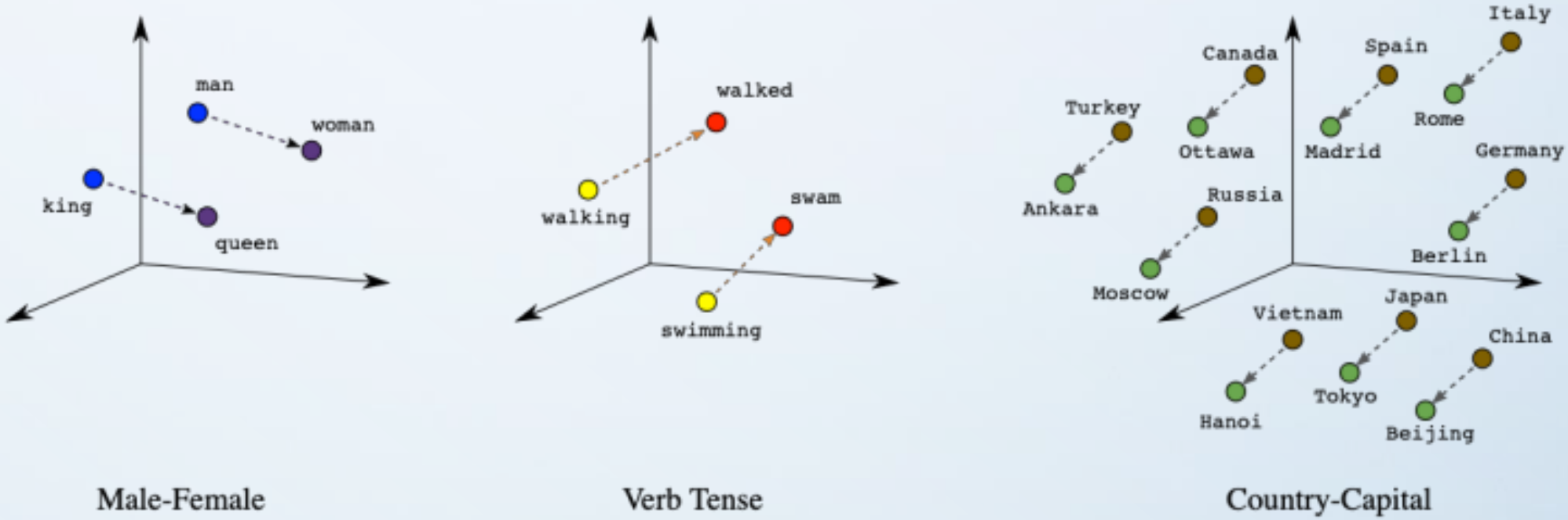
- Принцип работы:
  - *Модель word2vec принимает большой текстовый корпус в качестве входных данных и сопоставляет каждому слову вектор, выдавая координаты слов на выходе.*
  - *Сначала генерируется словарь корпуса, а затем вычисляются векторные представление слов, «обучаясь» на входных текстах.*
  - *Векторное представление основывается на контекстной близости: слова, встречающиеся в тексте рядом с одинаковыми словами (а следовательно, имеющие схожий смысл), будут иметь близкие (по косинусному расстоянию) векторы. Полученные векторные представления слов могут быть использованы для обработки естественного языка и машинного обучения.*
  - *Word2vec выполняет прогнозирование на основании контекстной близости слов. Так как инструмент word2vec основан на обучении нейронной сети, чтобы добиться его наиболее эффективной работы, необходимо использовать большие корпусы для обучения, что позволяет повысить качество предсказаний.*
- Источники:
  - Х. Лейн, Х. Хапке, К. Ховард. Обработка естественного языка в действии. — СПб.: Питер, 2020.
  - статья на русском языке,
  - оригинальная статья Т.Миколова,
  - пример визуализации.

Модель word2vec – алгоритмы обучения



- *Модель word2vec использует две возможные архитектуры:*
  - CBoW — архитектура, которая предсказывает текущее слово, исходя из окружающего его контекста.
  - Архитектура Skip-gram действует наоборот: она использует текущее слово, чтобы предсказывать окружающие его слова.
  - Построение модели word2vec возможно с помощью двух данных алгоритмов. Порядок слов контекста не оказывает влияния на результат ни в одном из этих алгоритмов.
  - Архитектура Skip-gram *хорошо работает для маленьких корпусов и редко встречающихся термов.* Архитектура CBoW *демонстрирует более высокую точность для часто встречающихся слов, и обучение занимает гораздо меньше времени.*

Модель word2vec – вычисления с векторами



Male-Female Verb Tense Country-Capital

Источник: https://www.baeldung.com/cs/convert-word-to-vector

- Получаемые на выходе векторные представления слов позволяют вычислять «семантическое расстояние» между словами. Так, можно находить похожие по значению слова.
- Пример: король относится к мужчине также, как королева к женщине.

Модель GloVe – описание

- Исследователи из Стэнфорда решили выяснить, почему Word2vec так хорошо работает, и найти оптимизируемую функцию стоимости. Они начали с подсчета совместных вхождений слов и занесения их в квадратную матрицу. Оказалось, что можно вычислить сингулярное разложение этой матрицы совместной встречаемости, разбив ее на те же две матрицы весов, что генерирует word2vec.
- Но в некоторых случаях модель Word2vec не сходится к тому глобальному минимуму, который получался у стэнфордских исследователей с помощью SVD. Именно от непосредственной оптимизации глобальных векторов (global vectors) совместной встречаемости слов (совместных вхождений в рамках всего корпуса) и получил свое название метод GloVe.
- Метод GloVe может формировать матрицы, эквивалентные входным и выходным матрицам весов Word2vec, в результате чего получается языковая модель с той же точностью, что Word2vec, но за намного меньшее время.
- Полученные представления отражают важные линейные подструктуры векторного пространства слов: получается связать вместе разные спутники одной планеты или почтовый код города с его названием.
- GloVe учитывает совместную встречаемость, а не полагается только на контекстную статистику. Векторы слов группируются вместе на основе их глобальной схожести.
- GloVe использует простую архитектуру без нейронной сети, поэтому векторные представления строятся быстрее, чем в случае word2vec. GloVe опережает word2vec на большинстве бенчмарков.
- Источники:
  - оригинальная статья

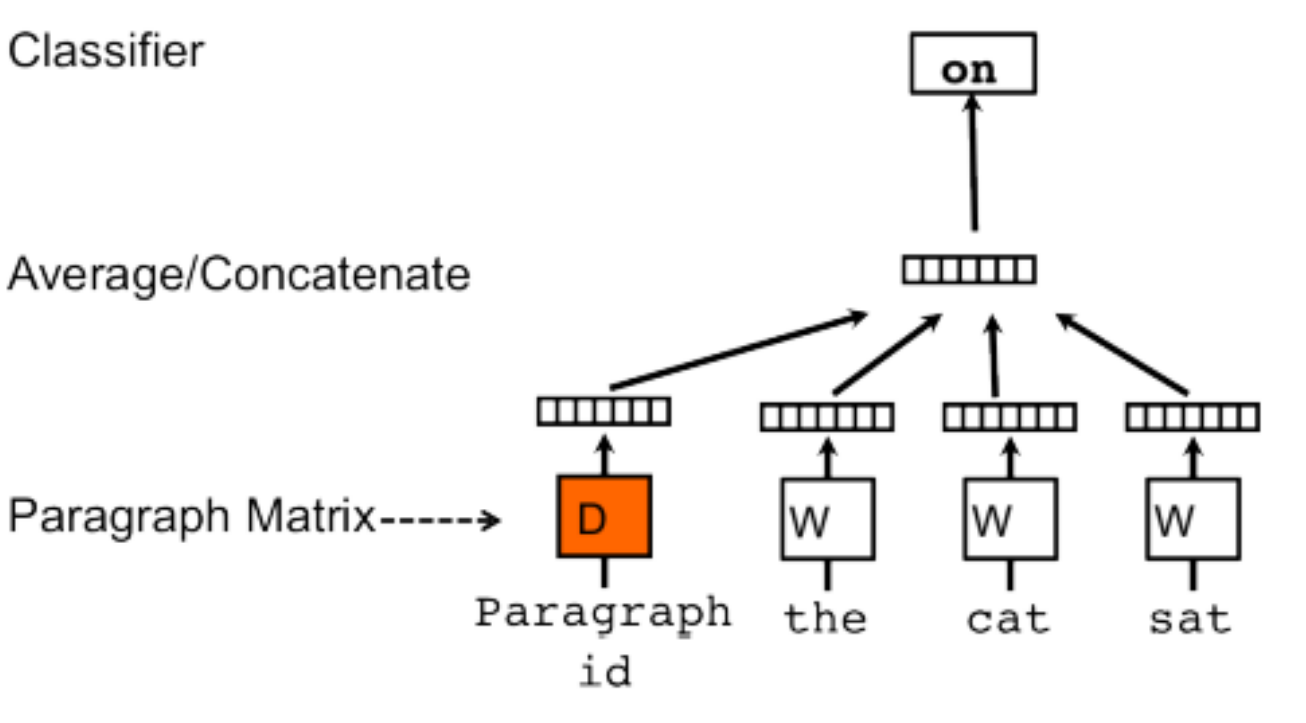
Модель fastText – описание

- Основной проблемой Word2vec и GloVe является необходимость формирования фиксированного словаря слов перед началом работы модели. Появление новых слов, отсутствующих в словаре, является проблемой.
- Модель fastText – это расширение модели Word2Vec, предложенное авторами этой модели.
- К основной модели Word2Vec добавлена модель символьных n-грамм. Каждое слово представляется композицией нескольких последовательностей символов определённой длины. Например, слово they в зависимости от гиперпараметров может состоять из "th", "he", "ey", "the", "hey". Вектор слова – это сумма всех его n-грамм.
- Поскольку количество n-грамм может быть очень велико, то n-граммы хэшируются с использованием хэш-функции Fowler–Noll–Vo.
- Далее могут использоваться подходы CBoW/Skip-gram для создания векторных представлений.
- Источники:
  - Официальный сайт,
  - оригинальная статья Т. Миколова,
  - Модели GloVe и fastText как улучшения Word2Vec – статья,
  - FastText: рецепт работы по коду – статья.
- *Пример использования моделей word2vec и fastText.*

Модель doc2vec – описание

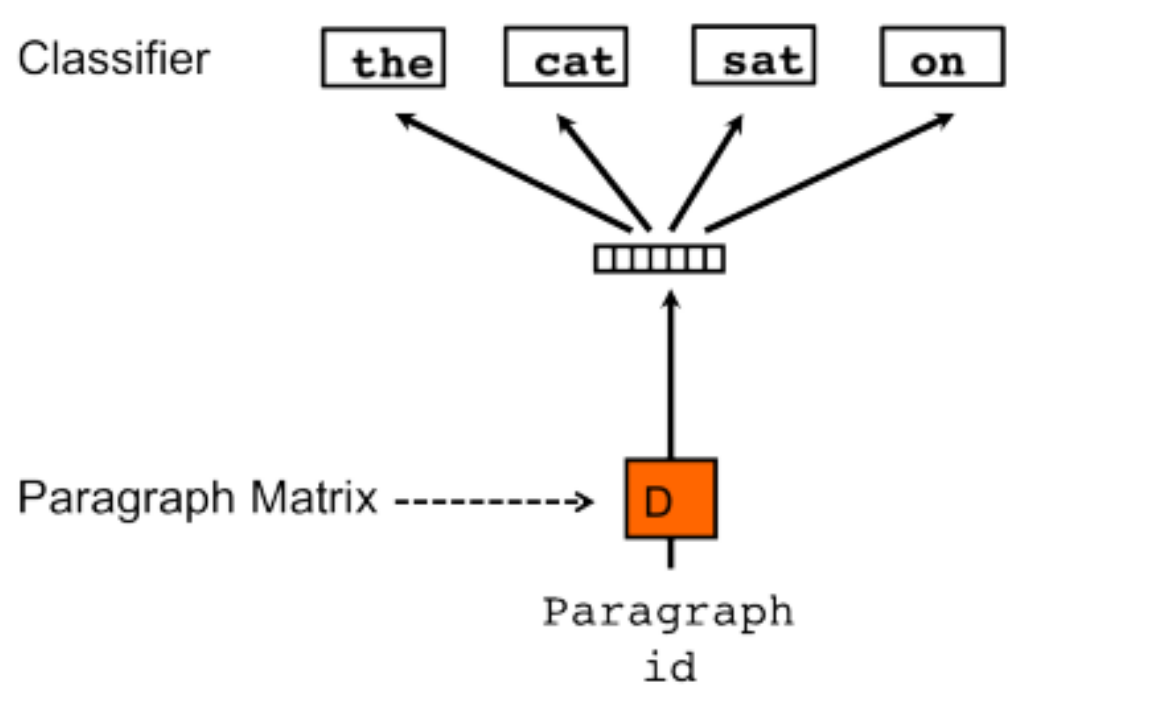
- По аналогии с векторными представлениями для слов, можно строить векторные представления для документов. В этом случае учитывается смысл не только отдельных слов, но и абзацев, предложений, документов.
- Модель представляет собой расширение модели Word2Vec, в которую дополнительно введен идентификатор параграфа - Paragraph Id (может использоваться также идентификатор документа или фрагмента документа).
- За счет расширения идеи Word2vec с помощью дополнительного вектора документа или параграфа, применяемого для предсказания слов, можно использовать полученный в результате обучения вектор документа для различных целей, например для поиска в корпусе схожих документов.
- Источники:
  - статья с описанием модели,
  - оригинальная статья авторов.

Модель doc2vec – алгоритмы обучения – PV-DM



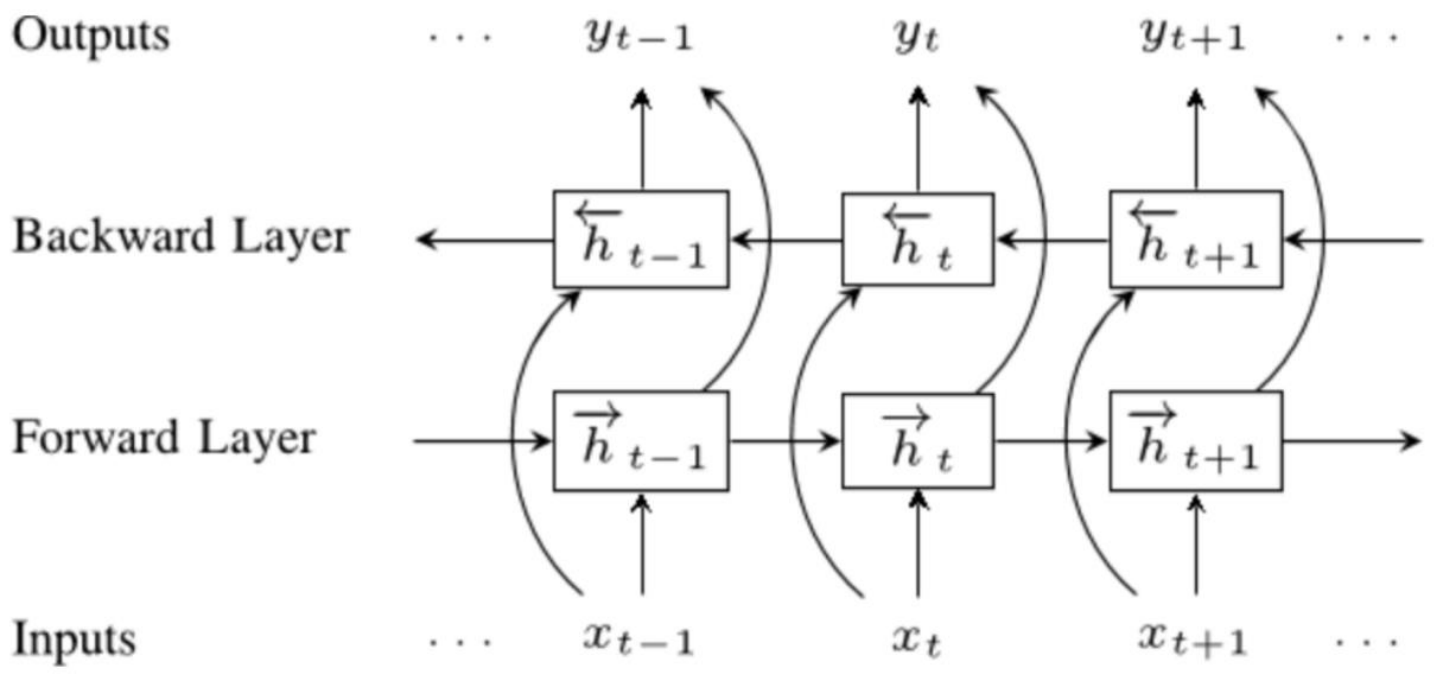
- В качестве расширения архитектуры CBoW используется архитектура PV-DM (Distributed Memory version of Paragraph Vector) — архитектура, которая предсказывает текущее слово, исходя из окружающего его контекста с учетом идентификатора параграфа.

Модель doc2vec – алгоритмы обучения – PV-DBOW



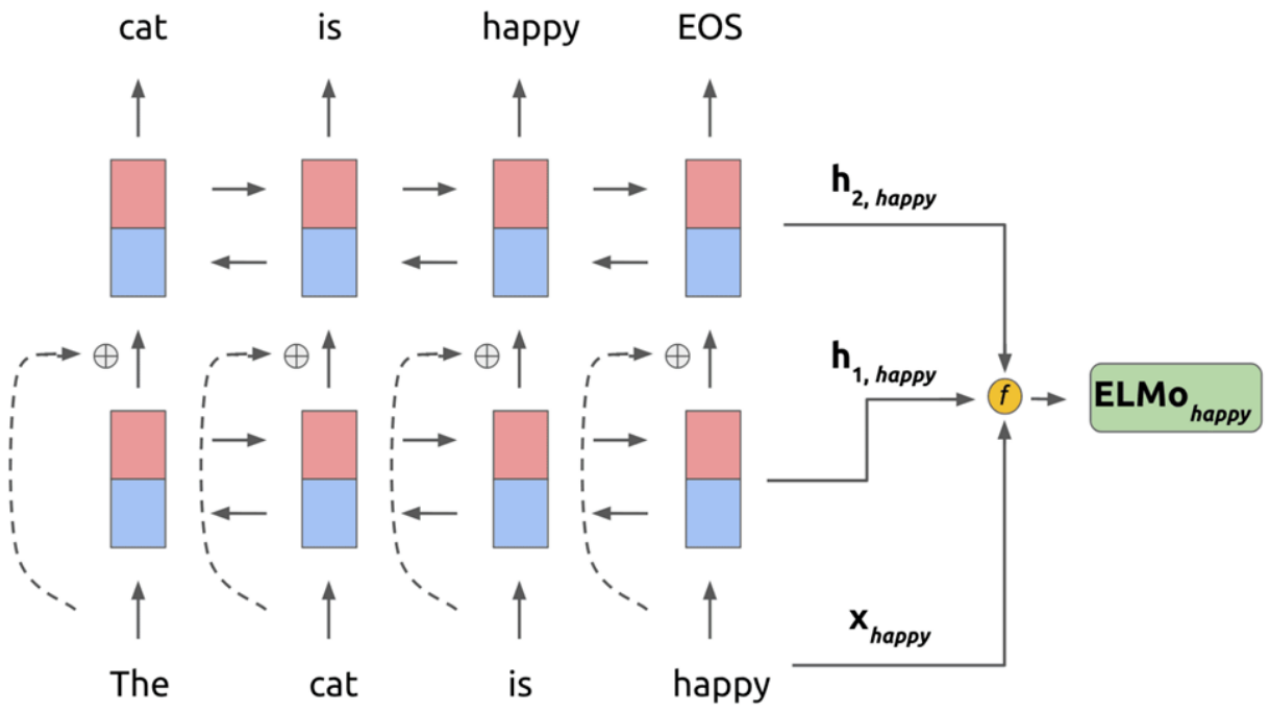
- Аналогом архитектуры Skip-gram является архитектура PV-DBOW (Distributed Bag of Words version of Paragraph Vector), которая предсказывает текст параграфа по его идентификатору.

Модель ELMo – описание и структура сети



- Строит векторные представления слов с учетом их встречаемости в контексте документа. Для этого используется двунаправленная LSTM-сеть (с прямым и обратным слоем), итоговый вектор слова является объединением векторов в каждой из сетей.
- Источники:
  - оригинальная статья авторов,
  - описание на paperswithcode,
  - статья с пояснениями.

Модель ELMo – пример



- Пример формирования векторного представления слова показан на рисунке.
- Преимущества модели:
  - Векторные представления слов строятся с учетом как их синтаксиса, так и семантики.
  - Модель принимает во внимание контекст слова (в том числе с учетом полисемии).
- Примеры использования в Python: первый, второй, третий.