ФЕДЕРАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ЦЕНТР "ИНФОРМАТИКА И УПРАВЛЕНИЕ" РОССИЙСКОЙ АКАДЕМИИ НАУК (ФИЦ ИУ РАН)

На правах рукописи
УДК 004.852

Кудинов Михаил Сергеевич

СТАТИСТИЧЕСКОЕ МОДЕЛИРОВАНИЕ РУССКОГО ЯЗЫКА С ПОМОЩЬЮ НЕЙРОННЫХ СЕТЕЙ

Специальность 05.13.17 —
«Теоретические основы информатики»

Диссертация на соискание учёной степени
кандидата технических наук

Научный руководитель:
ведущий научный сотрудник,
кандидат физико-математических наук
Чучупал В.Я.

Москва — 2016

Содержание

Введение

Задача статистического моделирования языка состоит в определении вероятностного распределения над цепочками слов в некотором языке. Данная задача естественным образом возникает в таких практических областях как распознавание речи, оптическое распознавание символов (OCR), распознавание рукописного текста, машинный перевод, проверка орфографии, предикативный ввод и других.

За минувшие 25 лет спрос на программные решения, связанные с обработкой текста, уже неоднократно переживал периоды роста, связанные сначала с появлением персональных компьютеров, затем со стремительным развитием интернета, и, наконец, с одновременным взрывным ростом социальных сетей и рынка мобильных устройств. При этом естественный язык остается важнейшим способом коммуникации, будь то ввод поискового запроса на миниатюрном экране мобильного телефона, подсказки автомобильного навигатора или бизнес-переписка. Практически во всех таких приложениях так или иначе используется языковая модель. Так, для удобного ввода текстов на мобильном телефоне, необходимо использовать системы предиктивного ввода, что практически сводится к прямому применению языковой модели; языковая модель — неотъемлемая часть систем распознавания речи, в том числе и в голосовом поиске; языковые модели используются в системах машинного перевода, качество которых на настоящий момент еще далеко от идеального, но все же неуклонно растет.

Историю языкового моделирования принято возводить к работам Шеннона, однако настоящая популярность статистических методов обработки текста началась лишь в 1980-е, с первыми успехами, полученными инженерами компании IBM. За сравнительно недолгую историю существования проблемы статистического моделирования языка было предложено большое количество различных подходов к ее решению, главным из которых по сей день остается подход, основанный на сглаженных n-граммных моделях. Было экспериментально продемонстрировано, что преимущества более сложных моделей в целом исчезают с ростом обучающей выборки. С начала 2010-х стал стремительно развиваться подход, основанный на рекуррентных нейронных сетях, описанный Т.Миколовым. В его работах было продемонстрировано, что с ростом объема обучающего корпуса преимущества предложенной им нейросетевой модели только увеличиваются. Рекуррентные модели за прошедшие годы нашли применение в различных областях от диалоговых систем до генерации текста по изображению.

Попытка прямолинейного применения модели Миколова к флективным языкам сталкивается с проблемой разреженности данных и большой вычислительной сложности в связи со свободным порядком слов и большим количеством грамматических форм, свойственным флективным языкам, в частности русскому. Одним из решений данной проблемы является раздельное предсказание начальных форм слова (лемм) и их морфологических форм. Данный подход позволил бы существенно снизить вычислительные затраты для обучения статистической модели русского языка.

Целью данной работы является построение эффективной статистической модели русского языка.

Для достижения поставленной цели необходимо было решить следующие задачи:

1. Аналитический обзор состояния проблемы и систематизация подходов к статистическому моделированию языка;
2. Экспериментальная оценка качества существующих языковых моделей для русского языка, выявление их недостатков и способов их устранения;
3. Разработка и теоретическое описание новых модификаций языковых моделей, устраняющих выявленные недостатки;
4. Разработка алгоритмов и программная реализация полученных моделей, а также экспериментальная проверка их эффективности.

Основные положения, выносимые на защиту:

1. Использование рекуррентной нейронной сети для статистического моделирования русского языка для предсказания начальных форм слова (лемм) более эффективно, чем n-граммная языковая модель, как с вычислительной точки зрения, так и с точки зрения качества предсказания;
2. Предсказание словоформ с помощью рекуррентной нейронной сети является неэффективным с вычислительной точки зрения;
3. Языковая модель, использующая отдельные классификаторы для предсказания лемм и морфологических признаков, требует значительно меньших вычислительных затрат;
4. Расширение рекуррентной модели на леммах за счет добавления признаков, полученных путем отображения левого контекста текущего слова в вектор действительных чисел, приводит к улучшению показателя перплексии новой модели по сравнению с исходной;
5. Предсказание морфологической формы, реализованное с помощью сверточной нейронной сети, задействующей морфологическую и лексическую информацию, снижает процент пословной ошибки при распознавании речи.

Научная новизна:

1. Впервые предложена статистическая языковая модель с раздельным предсказанием лемм и морфологических признаков, основанная на применении рекуррентной и сверточной нейронных сетей;

2. Разработана и реализована гибридная статистическая языковая модель на рекуррентной нейронной сети и тематическом разложении левого контекста, повышающая качество на более длинных текстах;

3. Впервые предложен, обоснован и экспериментально исследован метод расширения словаря языковой модели на рекуррентной нейронной сети за счет отдельной модели предсказания морфологической формы, основанной на сверточной нейронной сети.

В диссертации показана возможность построения статистической языковой модели, основанной на нейросетевом подходе, обеспечивающей эффективное предсказание морфологических форм и способной к учету дальних контекстных зависимостей между словами. Это определяет **теоретическую ценность** работы.

Использование предложенной модели в системах распознавания речи и текстового ввода позволяет улучшить качество данных систем. Этим определяется **практическая ценность** работы. Работы автора нашли практическое применение в технологиях компании «Самсунг Электроникс Ко., Лтд». В частности, автором был получен патент на изобретение «Голосовая связь на естественном языке между человеком и устройством» (RU 2583150).

Степень достоверности полученных результатов обеспечивается сходимостью теоретических оценок и экспериментальными результатами.

Работа проходила **апробацию** Основные результаты работы докладывались на: конференциях «Диалог–2014» (Бекасово, 2014), «Диалог–2015» (Москва, 2015), «SPECOM-2015», (Афины, 2015).

Все теоретические и экспериментальные результаты получены автором **лично**.

Публикации. Основные результаты по теме диссертации изложены в 3 печатных публикациях [1–3], 1 из которых издана в журнале, рекомендованном ВАК [1], 1 — в журнале, индексируемом базой SCOPUS [2], 1 — в тезисах докладов [3].

Объем и структура работы. Диссертация состоит из введения, четырех глав, заключения и библиографического списка использованных источников. Полный объем диссертации составляет 106 страниц с 22 рисунками и 12 таблицами. Список литературы содержит 115 наименований.

Первая глава **«Задача статистического моделирования языка»** является обзорной. В этой главе дается краткое описание задачи статистического моделирования языка, описываются основные метрики и вводятся базовые определения. Далее в главе описаны основные подходы к решению данной задачи.

Вторая глава **«Моделирование языка с помощью рекуррентных нейронных сетей»** посвящена анализу статистической языковой модели на основе рекуррентных нейронных сетей; проблемам данной модели, связанным с моделированием дальних контекстных зависимостей, и сложностям моделирования языков с богатой морфологией. Также в данной главе приводятся данные экспериментов по сравнению моделей предсказания лемм, основанных на рекуррентных нейронных сетях и сглаженных n-граммных моделях.

Третья глава **«Расширение рекуррентной архитектуры с помощью тематического моделирования»** посвящена построению улучшенной языковой модели на основе рекуррентной нейронной сети, направленной на уменьшение эффекта угасания градиента, ведущего к сложностям моделирования дальних контекстных зависимостей. Улучшение обеспечивается за счет использования векторного вложения левого контекста целевого слова, основанного на аппарате вероятностного тематического моделирования. В главе приводятся данные эксперимента, подтверждающего, что данная модель имеет меньшую перплексию по сравнению с исходной.

Четвертая глава **«Предсказание морфологических характеристик с помощью нейронной сети»** посвящена описанию модели предсказания морфологической формы слова с помощью сверточной нейронной сети. В главе приводятся данные эксперимента, подтверждающего, что использование модели морфологии ведет к снижению уровня пословной ошибки.

Заключение содержит основные результаты диссертационной работы, а также направления дальнейших исследований.

Глава 1

Задача статистического моделирования языка

1.1 Введение

Глава 1 посвящена обзору методов статистического моделирования языка. Во 2-ом разделе дается формальная постановка задачи статистического моделирования языка и ранжирования гипотез распознавания речи. В разделе 3 описаны методы оценки качества статистической модели, проанализированы их преимущества и недостатки. В разделе 4 приведен обзор методов статистического моделирования языка. В разделе 5 обсуждается применение моделей языка, основанных на лингвистической информации. В разделе 6 речь идет об эффективных языковых моделях, основанных на искусственных нейронных сетях. Наконец, в заключительном 7-ом разделе содержатся основные выводы, касающиеся решения задачи статистического моделирования естественного языка.

1.2 Постановка задачи статистического моделирования языка

1.2.1 Вероятностная формулировка

Неформально говоря, целью статистического моделирования языка является различение возможных (вероятных) или невозможных (маловероятных) цепочек слов в данном языке. Данная задача естественным образом возникает в таких практических областях как распознавание речи, оптическое распознавание символов (OCR), распознавание рукописного текста [4], машинный перевод [5], проверка орфографии, предикативный ввод и других. В последнем случае задача предстает в чистом виде, т.е. требуется предсказать следующее слово, при условии уже известного левого контекста.

Рассмотрим последнюю постановку формально.

Пусть требуется оценить вероятность появления последовательности слов w_1^t в языке L.

$$\mathsf{P}(w_1,\ldots,w_t) = \mathsf{P}(w_1,\ldots,w_{t-1})\,\mathsf{P}(w_t|w_1,\ldots,w_{t-1}) = \prod_{i=1}^{t}\mathsf{P}(w_i|w_1,\ldots,w_{i-1})$$

Использование данной модели в чистом виде, очевидно, потребовало бы оценки вероятностей $\mathsf{P}(w_i|w_1,\ldots,w_{i-1})$ для всех допустимых последовательностей слов в качестве параметров, что неосуществимо на практике. Поэтому на последовательностях вводится определенный класс эквивалентности Cl: т.е. все последовательности, попадающие в класс Cl представляются эквивалентными в данной статистической модели [6]

$$\mathsf{P}(w_t|w_1\ldots w_{t-1}) = \mathsf{P}(w_t|Cl(w_1\ldots w_{t-1}))$$

Выбор в качестве класса эквивалентности совпадения последних $n-1$ слов последовательности дает в результате широко известные n-граммные модели:

$$\mathsf{P}(w_t|w_1\ldots w_{t-1}) = \mathsf{P}(w_t|w_{t-n+1}\ldots w_{t-1}) \tag{1.1}$$

Оценка вероятности $\mathsf{P}(w_t|w_1\ldots w_{t-1})$ по методу максимального правдоподобия приводит к следующей очевидной формуле:

$$\mathsf{P}(w_t|w_1,\ldots,w_{t-1}) = \frac{C(w_1,\ldots,w_t)}{C(w_1,\ldots,w_{t-1})}, \tag{1.2}$$

где $C(w_1,\ldots,w_t)$ — количество появлений последовательности $w_1,\ldots,w_t$ в обучающей выборке [7]. При $n=1$ (*униграммная модель*) вероятности $p(w_t)$ соответствуют частотам слов $w \in V$ в корпусе.

Таким образом, для n-граммной модели необходимо оценить $|V|^n$ параметров, где $|V|$ — размер словаря, т.е. количество различных словоформ в обучающей выборке. Так, для словаря объемом $|V| = 20000$ словоформ в рамках биграммной модели пришлось бы оценить $|V|^2 = 4\cdot 10^8$ параметров. Следовательно, на достаточно большом корпусе объемом в 10 миллионов словоупотреблений может быть оценено не более 2.5% от предполагаемого объема модели. Остальные биграммы получат нулевую вероятность. Очевидно, что с увеличением длины n-граммы ситуация будет осложняться. С другой стороны, более длинные n-граммы позволяют получить лучшую предсказательную модель [6]. Данное наблюдение является частным случаем *проклятья размерности* — проблемы, широко известной в машинном обучении [8].

Данная проблема напрямую сказывается на результатах работы системы распознавания речи. В классическом подходе результат системы распознавания определяется как

$$\hat{W} = \arg\max_W \mathsf{P}(W)\,\mathsf{P}(A|W),$$

где W — множество последовательностей слов в данном языке, A — последовательность векторов акустических признаков [7].

Очевидно, что приписывание нулевой вероятности истинной последовательности слов W_0 автоматически ведет к неверному результату вне зависимости от того, насколько четким было произнесение.

Таким образом, задачей статистического моделирования языка является оценка вероятности последовательностей слов в данном языке, причем никакая последовательность не должна получить нулевую вероятность.

1.2.2 Использование языковых моделей в распознавании речи

Стандартным подходом к интеграции языковой модели в алгоритм распознавания речи является использование *лучевого поиска Витерби* (*Viterbi beam search*). Лучевой поиск Витерби представляет собой поиск в ширину в графе состояний скрытых марковских моделей. Граф поиска строится динамически путем добавления последовательностей аллофонов, соответствующих гипотетическим словам. Цепочка моделей аллофонов формирует модель слова. Вероятности переходов между моделями аллофонов внутри моделей слов принимаются равными 1. Конкатенация моделей слов формирует модель предложений. В отличие от переходов внутри моделей слов переходы вероятности межсловных переходов принимают значения, задаваемые языковой моделью n-го порядка [9].

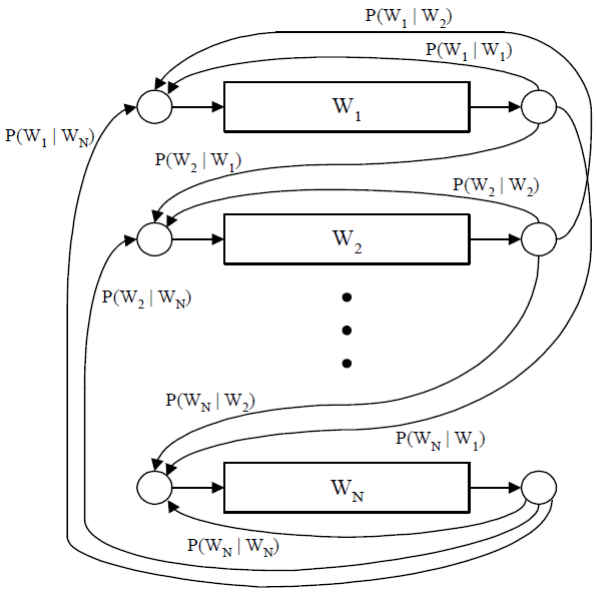

Рисунок 1.1: Порождение гипотез в графе поиска в форме конечного автомата [9]

Как только цепочка аллофонов, соответствующая слову w доходит до финального состояния, среди всех цепочек, заканчивающихся в w, выбирается наилучшая. Опуская неважные для дальнейшего изложения детали, алгоритм можно резюмировать следующим образом [9]: В алгоритме выше V — словарь, $h(w)$ гипотетическая цепочка $w, \ldots, w_{t-1}$, предшествующая w, :: обозначает конкатенацию.

Algorithm 1.2.1 Распознавание слитной речи

Вход: X; // *Векторы акустических признаков* $1 \dots T$
для $t = 1, \dots, T$
для всех $w \in V$
продолжить сопоставление $X_{start(w)}^{t-1}$ до момента t используя лучевой поиск Витерби
для всех $w \in V$
если слово w закончилось **то**
$\mathsf{P}(h(w),w) = \min_v[\log(\mathsf{P}(w|v) + \mathsf{P}(X_{start(w)}^{t}|w)]$
$h(w) = h(w) :: \arg\min_v[\log(\mathsf{P}(w|v) + \mathsf{P}(X_{start(w)}^{t}|w)]$

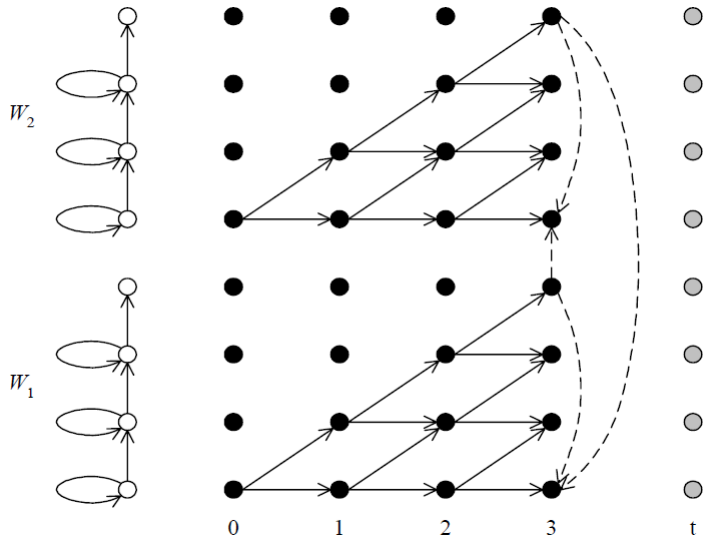

Рисунок 1.2: Развернутый граф поиска. Сплошные линии соответствуют внутрисловным переходам: их вероятности либо задаются моделью аллофонов, либо равны 1. Пунктирные линии имеют веса соответствующие вероятностям переходов между словами [10]

Поскольку переходы между состояниями внутри слов жестко заданы последовательностью фонем, граф поиска верной гипотезы в распознавании слитной речи можно представить как граф, узлами которого являются слова из словаря V, а взвешенными ребрами — возможные переходы между ними. Весами ребер, соответственно служат вероятности, заданные языковой моделью. При использовании биграммной языковой модели (см. пункт 1.4) достаточно помнить лишь вероятности переходов из предыдущего слова v в текущее слово w. Соответственно для шага максимизации достаточно держать в оперативной памяти $|V|$ гипотез. Использование триграммной модели требует уже $|V|^2$ гипотез. Очевидно, что с ростом порядка модели, количество поддерживаемых гипотез растет экспоненциально, поэтому на практике чаще всего используется поиск в несколько проходов. На первом проходе используется сравнительно простая n-граммная модель второго или третьего порядка. Поиск выдает n лучших гипотез в виде списка или сети (*latice*). Последующая обработка более сложными языковыми моделями фактически является задачей ранжирования гипотез. На этапе ранжирования могут быть применены сложные и более требовательные к ресурсам языковые модели, способные существенно повысить качество распознавания речи [11].

1.3 Методы оценки качества языковой модели

1.3.1 Энтропия и перплексия

Оценка качества статистической модели языка, предназначенной для системы распознавания речи, производится либо измерением *перплексии* на тестовой выборке, либо непосредственным измерением изменения уровня пословной ошибки в эксперименте по распознаванию [6]. Перплексия тестовой последовательности $w_1, \ldots, w_N$ в модели θ определяется как:

$$PPL = \sqrt[N]{\frac{1}{\mathsf{P}(w_1, \ldots, w_N|\theta)}} = \sqrt[N]{\frac{1}{\prod_{i=1}^{N} \mathsf{P}(w_i|w_1, \ldots, w_{i-1}, \theta)}}$$

или в экспоненциальной форме:

$$PPL = 2^{-\frac{1}{N}\sum_{i=1}^{N} \log_2 \mathsf{P}(w_i|w_1,\ldots,w_{i-1},\theta)} \tag{1.3}$$

Выражение, стоящее в показателе степени в уравнении 1.3, называется кросс-энтропией и является эквивалентным показателем качества модели ([11]). Фактически кросс-энтропия является усредненным логарифмом вероятности, которую модель θ приписывает выборке, усредненный по ее длине, т.е. правдоподобием модели θ. Кроме того, из свойств кросс-энтропии следует, что модель, обеспечивающая наименьшую кросс-энтропию, наиболее близка к истинной модели источника [12].

Физический смысл перплексии менее очевиден. В [6] использование перплексии в качестве метрики объясняется тем, что исторически сложность поиска гипотезы в процессе распознавания речи оценивалась через коэффициент ветвления в графе поиска. Фактически перплексия является коэффициентом ветвления в графе поиска, где все гипотезы равновероятны. Известна положительная корреляция между перплексией и уровнем пословной ошибки.

В то же время, по-видимому, использование перплексии как основной метрики качества статистической модели объясняется большей «чувствительностью» (а значит, и более «внушительными» изменениями абсолютных значений) в сравнении с кросс-энтропией [13]

Таблица 1.1: Соответствие между снижением энтропии и перплексии в пределах 1 бита [11]

Изменение энтропии	.01	.1	.16	.2	.3	.4	.5	.75	1
Изменение перплексии	0.69%	6.7%	10%	13%	19%	24%	29%	41%	50%

Однако, в [11] приводится неформальное рассуждение о том, что пословная ошибка должна изменяться примерно линейно относительно энтропии, т.е. небольшие изменения перплексии не оказывают значительного влияния на качество распознавания. Более того, распространенная практика отчетов об относительном изменении перплексии часто приводит к

тому, что результаты слабо сопоставимы друг с другом. Так 30%-му изменение переплексии соответствует ее снижение и с 2 пунктов до 1.4, и с 2000 до 1400. В то время как соответствующее изменение энтропии в первом случае (с 2 бит до 0.49) составит 51%, а во втором — 4.7%. Оба этих факта свидетельствуют о том, что измерения энтропии являются боле корректными и должны присутствовать в отчетах:

$$H(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\log_2 \mathsf{P}(w_i|w_1,\ldots,w_{i-1},\theta)$$

И энтропия, и перплексия имеют существенный недостаток, состоящий в том, что их оценка предполагает, что при предсказании следующего слова модели доступен *истинный* левый контекст. В то время как для систем предикативного ввода это в общем верно, языковая модель, встроенная в систему распознавания, не обладает таким качеством, т.е. неверно распознанное слово оказывает разрушительное воздействие на дальнейшее предсказание. Эта проблема становится особенно очевидна в случае с *кэшевыми языковыми моделями*. В этом случае энтропия оказывается низкой, однако это не приводит к снижению уровня пословной ошибки. Именно поэтому энтропия и перплексия не могут использоваться сами по себе.

1.3.2 Уровень пословной ошибки

Рассмотрим теперь другую важную метрику — уровень пословной ошибки. Уровень пословной ошибки является основной мерой качества системы распознавания речи и определяется как:

$$WER = \frac{S + D + I}{N},$$

где S — количество замен в варианте распознавание относительно референсной транскрипции, D — количество удалений, I — количество вставок и N — общее число слов в референсной транскрипции [9]. При вычислении WER выбирается тройка $\{S,D,I\}$, минимизирующая числитель в смысле редакторского расстояния [14].

Таким образом, WER напрямую измеряет качество системы распознавания речи путем подсчета количества ошибок, совершенных системой. Основным недостатком WER, помимо очевидного неудобства, связанного с необходимостью иметь полную систему распознавания, является низкая воспроизводимость результатов. Фактически в силу того, что WER зависит от настроек конкретной системы распознавания, результаты невозможно воспроизвести на большинстве других систем. Более того, очевидно, что снижение WER с ростом качества языковой модели тем сильнее, чем хуже работает акустический компонент системы распознавания, поэтому оценка эффективности языковой модели опосредованно зависит от качества внешнего по отношению к ней акустического модуля. В то же время данный показатель является репрезентативным при сравнении языковых моделей на одной системе.

Известной проблемой «классического» WER является одинаковое штрафование важных полнозначных слов и служебных слов, несущих лишь грамматическую информацию. Поскольку специфика речи такова, что служебные слова часто не имеют ударения или вовсе «проглатываются», WER не вполне точно отражает качество анализируемой системы. Ухудшает ситуацию и большая частотность служебных слов. Наиболее подходящий для WER способ применения — это оценка систем для записи под диктовку. В то же время очевидно, что для ряда задач подобная точность не требуется [9].

Безусловно, при оценке WER не всегда есть гарантия правильности референсной транскрипции и полученная оценка может быть несколько завышена.

В заключение стоит отметить, что как уровень пословной ошибки, так и энтропия сильно зависят от используемых данных и выбранных *baseline*-систем, что еще больше усложняет сравнимость результатов [13].

1.4 Методы статистического моделирования языка

Начиная с работ [15], [16], вышедших в 80-х, было предложено немало различных методов статистического моделирования языка. В вышедшем в 2000 году масштабном исследовании [11] сотрудниками Microsoft Corp. на обширном (разумеется, для своего времени) корпусе английского языка был проведен анализ существовавших к тому времени техник. Исследование показало, что при достаточном объеме обучающих данных почти ни один из методов не дает улучшений по сравнению с классическими методами n-граммного сглаживания [16] и [17]. Поскольку со времени выхода [11] новых методов почти не появлялось, результаты могут считаться актуальными. Отдельные комментарии будут даны относительно успешности применения описываемых методов к флективным языкам.

1.4.1 Сглаженные n-граммные модели

Выше уже было отмечено, что применения одной только оценки максимального правдоподобия недостаточно для построения качественной языковой модели вследствие *проклятья размерности*. Стандартным средством борьбы с данной проблемой является внесение поправок в оценки распределения слов. Данная техника называется *сглаживанием*. Основными методами реализации сглаживания являются *интерполяция* и *отступление* (*backing-off*) [10].

Остановимся вкратце на общих принципах каждого из методов.

Оба метода основаны на *дисконтировании*, т.е. перераспределении вероятностной массы с более частотных событий на редкие и ненаблюдаемые. Реализовать дисконтирование можно простейшим образом путем интерполяции моделей разного порядка от 1 до n. Триграммная модель, сглаженная с помощью биграммной и униграммной моделей методом интерполяции

задается следующим равенством:

$$P_{interp}(w|w_{t-1}w_{t-2}) = \lambda \mathsf{P}(w|w_{t-1}w_{t-2}) + (1-\lambda)[(\mu \mathsf{P}(w_t|w_{t-1}) + (1-\mu)\mathsf{P}(w)], \tag{1.4}$$

где $0 \leq \lambda,\mu \leq 1$ — нормировочные коэффициенты.

Несмотря на свою простоту и неплохие результаты ([18]) простая интерполяция n-граммных моделей разного порядка сравнительно редко используется на практике. Более практичными подходами являются методы сглаживания Катца [16] и Кнесера-Нея [17].

Математически оба метода могут быть обоснованны как различные предельные случаи оценок вероятностей, получаемых при кросс-валидации [10]. Здесь приводится лишь краткое неформальное описание.

Сглаживание Катца основано на оценке Гуда-Тьюринга [10] для дисконтирования частот наблюдаемых событий в пользу ненаблюдаемых. Так, появление фразы *проржавленный околдовавшая нейросеть* в обучающем корпусе один раз может быть и, скорее всего, является случайным выбросом, и вероятность данного события может быть существенно меньше, чем $\frac{1}{N_{train}}$, где N_{train} — количество токенов в обучающей выборке. Гудом была получена оценка максимального правдоподобия для количества появления n-граммы в корпусе с использованием скользящего контроля по отдельным объектам:

$$r = (r+1)\frac{n_{r+1}}{n_r}, \tag{1.5}$$

где $n_r = |\{w_1,\ldots,w_n|C(w_1,\ldots,w_n) = r\}|$ — количество различных n-грамм, появившихся в корпусе ровно r раз. Например, из формулы (1.5) видно, что общая частота ненаблюдаемых событий оценивается как $\frac{n_1}{N_{train}}$.

Данная оценка лежит в основе сглаживания Катца:

$$P_{Katz}(w_t|w_{t-n+1},\ldots,w_t) = \begin{cases} \frac{C^*(w_{t-n+1},\ldots,w_t)}{C(w_{t-n+1},\ldots,w_{t-1})}, \text{если } C(w_{t-n+1},\ldots,w_t) > 0; \\ \alpha(w_{t-n+1},\ldots,w_{t-1}) \cdot P_{Katz}(w_t|w_{t-n+2},\ldots,w_{t-1}), \quad \text{в противном случае,} \end{cases} \tag{1.6}$$

где α— нормирующий коэффициент. Из приведенных формул видно, что в случае, когда вероятность $P(w_t|w_{t-n+1})$ может быть оценена по корпусу, частота события дисконтируется. Напротив, если невозможно оценить вероятность появления токена w, следующего за цепочкой токенов $w_{t-n+1},\ldots,w_{t-1}$, мы пытаемся получить эту вероятность, основываясь на более короткой истории $w_{t-n+2},\ldots,w_{t-1}$.

Использование сглаживания Катца широко распространено в практических приложениях. Тем не менее, модель Катца страдает как минимум от одной проблемы.

Из нижней формулы в определении сглаживания Катца видно, что для частотного токена w оценка вероятности n-граммы, отсутствующей в корпусе будет высокой вне зависимости от его сочетаемостных свойств. Так, токен *франциско* может иметь высокую частоту за счет

высокой частоты встречаемости цепочки *сан франциско* в корпусе. Соответственно оценка вероятности цепочки *из франциско* имеет шансы получить достаточно большую вероятность, что явно не соответствует ожиданиям.

Идея сглаживания Кнесера-Нея состоит в том, что при расчете *обобщенного распределения*, роль которого в сглаживании Катца играет частотное распределение с менее специфичным контекстом, используется количество различных левых контекстов, в которых встречается целевое слово:

$$P_{KN}(w_t|w_{t-n+1}\dots w_t) = \begin{cases} \frac{C(w_{t-n+1}\dots w_t)-D}{C(w_{t-n+1}\dots w_{t-1})}, \text{если } C(w_{t-n+1}\dots w_t) > 0; \\ \alpha(w_{t-n+1}\dots w_{t-1}) \cdot \frac{|\{w_{t-n+1}\dots w_{t-1}|C(w_{t-n+1}\dots w_t)>0\}|}{\sum_{w'}|\{w_{t-n+1}\dots w_{t-1}|C(w_{t-n+1}\dots w_{t-1}w')>0\}|}, \text{в противном} \\ \text{случае,} \end{cases} \tag{1.7}$$

где D — параметр дисконтирования, α — нормирующий коэффициент. Как показано в [11], данная версия сглаживания Кнесера-Нея наряду с другими моделями с отступлением, такими как сглаживание Катца, показывают слабые результаты для низкочастотных, т.е. встреченных менее трех раз, n-грамм. Эксперименты показали, что лучшей модификацией модели Кнесера-Нея является модель Кнесера-Нея с интерполяцией:

$$\begin{aligned} P_{KN}(w_t|w_{t-n+1}\dots w_t) = \frac{C(w_{t-n+1}\dots w_t) - D}{C(w_{t-n+1}\dots w_{t-1})} + \\ + \lambda(w_{t-n+1}\dots w_t)\frac{|\{w_k\dots w_{t-1}|C(w_{t-n+1}\dots w_t) > 0\}|}{\sum_{w'}|\{w_{t-n+1}\dots w_{t-1}|C(w_{t-n+1}\dots w_{t-1}w') > 0\}|} \end{aligned} \tag{1.8}$$

В той же серии экспериментов было показано, что все перечисленные модели сглаживания достигают минимума энтропии на тестовой выборке при порядке модели равном 4 или 5, после чего график зависимости энтропии от порядка модели выходит на плато. Данные диссертации [19], посвященной языковым моделям русского языка, также указывают на сложности, связанные с тренировкой моделей большего порядка для русского языка. Таким образом, в качестве базовой (*baseline*) модели будет использована 4-граммная модель Кнесера-Нея с интерполяцией (KN4).

1.4.2 N-граммные модели с пропуском

С ростом порядка n-граммы растет и разреженность данных. Таким образом, шансы встретить конкретную n-грамму понижаются. Вместе с тем, вполне вероятно, что мы можем встретить некоторую «похожую» n-грамму: например, *бесцветные мысли* вместо *бесцветные зеленые мысли*. Данная проблема стоит особенно остро в случае русского языка с его большим количеством морфологических форм и свободным порядком слов. Естественным

способом борьбы с разреженностью такого рода являются n-граммные модели с пропуском (*skip models*) [20], [21], [22].

Естественным способом использования модели с пропуском является ее интерполяция с моделями без пропуска. Например, для 5-граммной модели:

$$\mathsf{P}(w_t|w_{t-4}\dots w_{t-1}) = \lambda\,\mathsf{P}(w_t|w_{t-4}w_{t-3}w_{t-2}w_{t-1}) + \\ + \mu(\mathsf{P}(w_t|w_{t-4}w_{t-3}w_{t-1})) + (1-\mu-\lambda)\,\mathsf{P}(w_t|w_{t-4}w_{t-2}w_{t-1})$$

, где $\lambda + \mu < 1$. Либо «приближение» модели более высокого порядка:

$$\mathsf{P}(w_t|w_{t-3}\dots w_{t-1}) = \lambda\,\mathsf{P}(w_t|w_{t-2}w_{t-1}) + \mu\,\mathsf{P}(w_t|w_{t-3}w_{t-1}) + (1-\mu-\lambda)\,\mathsf{P}(w_t|w_{t-3}w_{t-2})$$

В ходе экспериментов [11] было показано, что 5-граммная модель с пропусками дает улучшение менее чем на 0.1 бит относительно KN5. Наибольшее улучшение наблюдается при размерах корпуса порядка 10^6 токенов, после чего различия с 5-граммной моделью постепенно стираются.

Более ощутимых улучшений удается достичь относительно триграммной модели. В данном случае, интерполируя предсказания, где предиктором являются пары слов, не стоящих рядом, удается получить улучшения порядка 0.12 бит, если брать все пары слов, находящихся на расстоянии не более 5 токенов. Таким образом, данная техника является полезной в отсутствие достаточно большого обучающего корпуса. С увеличением корпуса улучшения относительно модели со сглаживанием Кнесера-Нея исчезают.

1.4.3 Модели с классами

Другим естественным способом борьбы с разреженностью данных является объединение похожих по смыслу слов в семантические, лингвистические или просто частотные классы. Допустим следующую ситуацию. В корпусе мы встретили фразы *Телепередача выходит по четвергам* и *Свежий выпуск газеты выходит по средам*. Обученная модель со сглаживанием не присвоила бы высокой вероятности предложению *Телепередача выходит по пятницам*, хотя интуитивно это, очевидно, должно быть так. Распишем вероятность последней триграммы:

$$\mathsf{P}(\textit{пятницам}|\textit{выходит по}) = \mathsf{P}(\textit{ДЕНЬ\_НЕДЕЛИ}|\textit{выходит по})\times \\ \times\,\mathsf{P}(\textit{пятницам}|\textit{выходит по ДЕНЬ\_НЕДЕЛИ}) \quad (1.9)$$

Тогда вероятность $\mathsf{P}(\textit{пятницам}|\textit{выходит по ДЕНЬ\_НЕДЕЛИ})$ последовательности, отсутствующей в обучающей выборке, мы можем оценить с помощью сглаживания, например как $\mathsf{P}(\textit{пятницам}|\textit{ДЕНЬ\_НЕДЕЛИ})$ и подставить в исходное выражение.

Существует множество вариантов модели (1.9). Так, в [23] предлагается более последовательная кластерная модель без использования слов в качестве предикторов:

$$\mathsf{P}(w|w_{i-2}w_{i-1}) \approx \mathsf{P}(W|W_{i-2}W_{i-1}) \times \mathsf{P}(w|W)$$

Впрочем, далее авторы предлагают интерполировать данную модель с классической триграммной.

Наиболее эффективным типом моделей с классами, как показано в [11], являются модели, в которых вероятность n-граммы $w_1, \ldots, w_n$ разбивается в произведение вероятностей, согласно формуле:

$$\begin{aligned}\mathsf{P}(w|w_{i-2}w_{i-1}) = (\lambda\,\mathsf{P}(W|w_{i-2}w_{i-1}) + (1-\lambda)\,\mathsf{P}(W|W_{i-2}W_{i-1}))\times \\ \times\,(\mu\,\mathsf{P}(w|w_{i-2}w_{i-1}W) + (1-\mu)P(w|W_{i-2}W_{i-1}W))\end{aligned} \quad (1.10)$$

Таким образом, сомножители соответствуют вероятностям класса и токена, принадлежащего данному классу. Каждый сомножитель при этом вычисляется путем интерполяции моделей, основанных соответственно на токенах и на кластерах.

Использование кластеров оказывается довольно эффективным даже на больших обучающих множествах: в упомянутом исследовании Microsoft Corp. показано, что модель (1.10) позволяет понизить энтропию почти на 0.5 бита относительно триграммной модели со сглаживанием Катца и 0.4 бита относительно модели со сглаживанием Кнесера-Нея, причем улучшение наблюдается даже на очень большом корпусе на 10^8 токенов. С увеличением корпуса различия становятся все менее заметными, что выглядит логичным, учитывая, что модели с классами являются средством борьбы с эффектами разреженности данных.

В [13] упоминается о том, что модели с классами демонстрируют хорошие результаты в снижении уровня пословной ошибки.

Пожалуй, основной проблемой моделей с классами становится собственно кластеризация или классификация, когда части слов из словаря класс приписывается экспертами. С ростом словаря проблема отнесения слова к какому бы то ни было классу и числа самих слов значительно усложняется. Ее решением выглядит использование моделей, основанных на латентно-семантическом анализе [24], латентном размещении Дирихле [25] или тематических моделях [26].

Подобное дистрибутивное представление лексики, однако, сталкивается с другой проблемой: математическое обоснование эффективных методов сглаживания Катца и Кнесера–Нея основано на моделировании обучения с кросс-валидацией [10], и существенно опирается на дискретность числа появлений n-граммы или класса в корпусе. Приписывание каждому слову вектора вероятностей вхождения в тот или иной кластер (например, тематического профиля слова) потребовало бы от модели уточнения [11]. Тем не менее, подобные модели тести-

ровались и продемонстрировали обнадеживающие результаты в том числе для флективных языков [27], [28].

1.4.4 Модели с кэшированием

Помимо разреженности данных, другим очевидным недостатком n-граммных моделей является ограниченность контекста. Интуиция подсказывает, что если брать во внимание лишь некоторые слова, особо важные для понимания тематики текста, можно компенсировать эффекты короткого контекста. Модели, основанные на этой идее называются *кэширующими* (*caching*) [29], [30].

Существует два основных типа кэширующих моделей. Собственно кэширующая модель:

$$\mathsf{P}_{cache}(w_n|w_{n-M}\dots w_{n-1}) \sim \frac{1}{M}\sum_{m=1}^{M}\delta(w_n,w_{n-m}),$$

где $\delta(x,y)$ — функция Кронекера. Тривиальным образом, данная модель присваивает большую вероятность словам, которые чаще повторялись среди ближайших M слов. Результатом интерполяции кэширующей модели с n-граммной является n-граммная модель, в которой возрастает вероятность слов, присутствующих в более длинном левом контексте ($M \gg n$).

Вторым типом кэширующих моделей являются т.н. *триггерные модели* [31]. Триггерная модель учитывает зависимости между частотами различных слов:

$$\mathsf{P}_{trig}(w_n|w_{n-M}\dots w_{n-1}) \sim \frac{1}{Z}\sum_{m=1}^{M}\alpha(w_n,w_{n-m}), \tag{1.11}$$

где $\alpha(w_n,w_{n-m})$ — «вес» зависимости между предсказываемым словом w_n и словом из левого контекста w_{n-m}; Z — нормирующий коэффициент. Как правило, подобные модели являются лог-линейными [10].

Кэширующие модели часто приводятся как пример недостаточности энтропийного критерия для оценки качества языковой модели. Так в [11] показано, что кэширующие модели обеспечивают наилучшее уменьшение энтропии относительно триграммной модели со сглаживанием Кнесера-Нея. Для наилучшей комбинации кэширующих моделей выигрыш в энтропии может достигать 0.6 бит для корпуса объемом 10^5 токенов и 0.23 бит для корпуса объемом более 10^6 токенов. С другой стороны, подсчет энтропии предполагает знание правильного левого контекста, что не является типовой ситуацией в распознавании речи. Вполне возможна ситуация, когда использование кэширующей модели приводит к «закреплению» ошибки. Единственное неверно распознанное слово в результате ведет к последующему увеличению вероятности появления данного ошибочного распознавания. В результате кэширующие модели характеризуются меньшей корреляцией между понижением энтропии и уровня пословной ошибки.

1.4.5 Смеси предложений (sentence mixture models)

В [32], [33] была предложена модель, основанная на наблюдении, состоявшем в том, что в стандартном для обучения и тестирования английских моделей корпусе Wall Street Journal [34] имеется лишь несколько тематических типов предложений: новости финансовых рынков, политика и т.д. Предположительно, тематика предложения существенным образом влияет на предсказание слов. В модели [33] тип предложения является скрытой переменной. Соответственно, вероятность предложения может быть представлена как:

$$\mathsf{P}(w_1 \dots w_t) = \sum_{j=0}^{S} P(s_j) \prod_{i=1}^{N} \mathsf{P}(w_i | w_{i-n+1}, \dots, w_{i-1}, s_j)$$

Разбиение предложений на типы может быть произведено либо методами кластеризации [11], либо с помощью EM-алгоритма [33]. Результаты, изложенные в [11] и [33] не совпадают: так, в [11] достигнутый выигрыш в перплексии относительно триграммной модели со сглаживанием Кнесера-Нея (KN3) составляет 9% против 19% во втором исследовании. Снижение уровня пословной ошибки соответственно 1.3% и 3%. Тем не менее, с увеличением числа предложений до 64 модель достигает выигрыша в энтропии в 0.3 бита. Более интересным результатом является то, что с ростом объема обучающего корпуса различия с KN3 не стираются, как это было в случае со всеми моделями, описанными ранее.

Впрочем, очевидным минусом модели является ее сильная ориентация на конкретную тематику и даже хуже — конкретный корпус. Довольно трудно представить, что для системы распознавания речи с неограниченным словарем окажется достаточным некоторое разумное количество тематических типов предложений.

1.4.6 Модель максимальной энтропии

Модель максимальной энтропии [35] является одной из наиболее успешных не только в задаче статистического моделирования языка, но и в компьютерной лингвистике в целом [36]. По историческим причинам [36] моделью максимальной энтропии в исследованиях по компьютерной лингвистике называют логистическую регрессию, осуществляющую последовательную классификацию элементов входной строки. Фактически данная модель в более узкой сфере уже была введена в формуле (1.11) для моделирования дальних зависимостей между словами, однако на самом деле *триггерные слова* являются лишь одним типом *признаков*, используемых моделью максимальной энтропии.

В общем виде вероятность слова w_t в модели максимальной может быть представлена так:

$$\mathsf{P}(w_t | w_{t-n+1} \dots w_{t-1}) = \frac{\exp \sum_{i=1}^{|F|} \phi_i f_i(w_{t-n+1}, \dots, w_t)}{Z(w_{t-n+1}, \dots, w_t)}$$

, где F — множество бинарных признаков последовательности $w_1, \ldots, w_n$, $f_i \in \{0,1\}$; $\{\phi_i | \phi_i \in \mathbb{R}; \phi_i \geq 0\}$ — множество весов признаков;

$$Z = \sum_{w'_t \in V} \exp \sum_{i=1}^{|F|} \phi_i f_i(w_{t-n+1}, \ldots, w'_t)$$

— функция разбиения, гарантирующая суммируемость к 1. Благодаря тому, что признаком может быть практически любая предикатная функция на последовательности токенов, модель максимальной энтропии может успешно заменить собой не только n-граммную модель, но также и модели с кэшированием и классами. Это делает модель максимальной энтропии потенциально самой мощной из всех перечисленных. При этом ее несомненным плюсом является единообразное представление разнородных статистических свойств, учет которых в совокупности ранее требовал интерполяции различных моделей.

Оборотной стороной мощности модели максимальной энтропии является относительная сложность ее обучения. Модель максимальной энтропии обучается градиентными методами, что является достаточно дорогим процессом в сравнении с описанными ранее моделями, для которых решение обычно выводится в замкнутой форме.

Тот факт, что модель максимальной энтропии использует и моделирует те же статистические свойства, что и описанные ранее модели, приводит к тому, что в целом модель комбинирует все достоинства и недостатки описанных ранее моделей (в зависимости от выбранных признаков) и работает не намного лучше их интерполяции [11].

Вместе с тем, использование модели максимальной энтропии позволяет значительно упростить работу исследователя в области статистического моделирования языка, практически полностью сведя ее к дизайну признаков.

Необходимость в эвристическом выборе признаков исчезает при переходе от модели максимальной энтропии к языковым моделям на искусственных нейронных сетях.

1.4.7 Латентно-семантический анализ и тематическое моделирование

Латентно-семантический анализ и тематическое моделирование дали мощный толчок в развитии *распределенных* представлений языковых единиц в конце 90-х – начале 2000-х годов. В работах [24], [37] проводятся эксперименты по интеграции латентно-семантического анализа (LSA) или латентного размещения Дирихле (LDA) с языковыми моделями. Латентно-семантическое индексирование основано на сингулярном разложении матрицы термин-документ с последующим понижением размерности. Вкратце алгоритм можно описать следующим образом:

1. Строится матрица $W_{V \times D}$ «термин–документ», где $w_{i,j}$, $1 \leq i \leq V$, $1 \leq j \leq D$, — количество появлений термина i в документе j.

2. Вычисляется SVD-разложение матрицы W:

$$W = USV^T$$

3. Вычисляется $\hat{W}$ приближение матрицы W матрицей ранга $R < R_W$:

$$\hat{W} = U_{V\times R} S_{R\times R} V^T_{R\times D}$$

4. Для вектора, не содержащегося в обучающем множестве, проекция в полученное семантическое пространство вычисляется как:

$$v = d^T U, \tag{1.12}$$

где d – исходный V-мерный вектор, v – преобразованный R-мерный вектор.

В работе [24] предлагается способ использования семантического представления в предсказании слов по контексту, в рамках которого левый контекст слова w_t рассматривается как «псевдодокумент», на основании которого строится условное распределение:

$$\hat{\mathsf{P}}(w_t|w_{t-n+1},\ldots,w_{t-1}) = \mathsf{P}(w_t|d_{t-1}),$$

где d_{t-1} – «псевдодокумент», r-мерная проекция вектора частот слов $w_1 \ldots w_{t-1}$ в *семантическое пространство*. Далее на основании корреляции между вложением слова w_t u и документа d_{t-1} v вычисляется вероятность предсказания [38]

$$\mathsf{P}(w_t|d_{t-1}) = \frac{\cos(uS^{1/2}, dS^{1/2}) - \min_W \cos(wS^{1/2}, dS^{1/2})}{\sum_l \cos(lS^{1/2}, dS^{1/2}) - \min_W \cos(wS^{1/2}, dS^{1/2})} \tag{1.13}$$

Предсказания, основанные на вероятностных тематических моделях устроены сходным образом. Основным отличием от LSA является использование неотрицательных матричных разложений взамен сингулярного. Поскольку данная задача является некорректно поставленной, на решение накладываются дополнительные ограничения двух типов: первый тип требует, чтобы элементы матрицы удовлетворяли вероятностным свойствам; второй тип ограничений определяется регуляризационными членами и задает различные тонкие настройки тематической модели [39]. Обучение таких моделей, как правило производится с помощью EM-алгоритма. К вероятностным тематическим моделям относятся PLSA (probabilistic latent semantic analysis — вероятностный латентно-семантический анализ) и LDA (latent Dirichlet allocation — латентное размещение Дирихле).

В [24] сказано, что использование LSA привело к 20% снижению перплексии и 9% снижению пословной ошибки относительно триграммной модели. Измерения перплексии проводились на корпусе объемом $42 \cdot 10^6$ токенов. Таким образом, результаты можно признать

хорошими. Тем не менее, логично предположить, что использование модели LSA так, как это описано выше, должно привести к тем же проблемам, что и в ситуации с моделью с кэшированием, однако результат будет более устойчив к случайным выбросам.

Очевидно, что можно использовать и более простую схему. Например, использовать тематический профиль слов в качестве признаков в модели максимальной энтропии, либо использовать в качестве таких признаков не вектор v, вычисленный по формуле (1.12), а среднее значение тематических профилей слов в левом контексте. В статье [28], где тематический профиль используется в качестве дополнительного вектора параметров нейронной сети, сходная схема используется для ускорения модели и показывает сопоставимые результаты.

1.5 Лингвистически мотивированные модели

1.5.1 Модели, использующие морфологическую информацию

За время исследований в области статистического моделирования языка было также предложено немало т.н. *лингвистически мотивированных* моделей. Под лингвистической мотивированностью в данном случае стоит понимать то, что используемая статистическая модель в той или иной степени основывается на лингвистической теории. Большинство из известных грамматических теорий основываются на гипотезе о существовании абстрактных языковых структур [40], [41] — например, синтаксической. Таким образом, наиболее эффективная из языковых моделей, n-граммная, лишена лингвистического обоснования. Есть, однако, и более практические соображения, приводящие исследователя в сторону линвистических моделей, и они непосредственно связаны с моделированием «сложных» языков с богатой морфологией — флективных (*русский*, *чешский*) или агглютинативных (*турецкий*, *финский*). С одной стороны, богатая морфология коррелирует со свободным порядком слов в предложении [42], с другой — повышает разреженность данных и пропорционально увеличивает размер словаря [19].

В основе структурных статистических моделей, как правило, лежат лингвистические модели морфологии и синтаксиса целевого языка [43].

Самым простым способом интеграции лингвистических знаний в языковую модель является использование информации о морфологических классах. Использование морфологической информации может понизить энтропию n-граммной модели за счет запоминания типовых фразовых конструкций без привлечения более сложных синтаксических моделей.

$$\mathsf{P}(w_t|w_{t-n+1},\ldots,w_{t-1}) = \lambda\,\mathsf{P}_{IKN}(w_i|w_{i-n+1},\ldots,w_{i-1}) + \\ + (1-\lambda)\,\mathsf{P}(POS(w_i)|POS(w_{i-n+1}),\ldots,POS(w_{i-1})) \quad (1.14)$$

В [44] интерполяция триграммной модели, предсказывающей частеречный (POS) тег привело к снижению энтропии на 0.14 бит и 1.1% снижению WER, что нельзя назвать выдающимся результатом. При этом объем используемого корпуса был менее 10^6 токенов. Учитывая опыт тестов из [11], можно предположить, что с дальнейшим увеличением объема обучающего корпуса разница в показателях модели исчезнет, а вес морфологической модели при ее интерполяции с триграммной — падать.

По большому счету, по крайней мере для английского языка, нет особых оснований ожидать значительного улучшения качества от морфологической модели. Информация о типовых последовательностях частей речи имплицитно содержится в n-граммах, и единственной причиной для привлечения морфологической модели может быть то, что морфология языка способствует усилению разреженности данных. Это в целом верно для флективных языков, однако, скорее, неверно для английского или китайского.

Минусом модели, основанной на морфологических классах, является еще и большая вычислительная сложность: фактически задача морфологического разбора есть дополнительная задача распознавания, которую потенциально необходимо решить для всех гипотез, возвращенных системой распознавания речи.

Другим примером использования морфологической информации является непосредственное моделирование распределения подслов: морфем и других элементов — *квазиморфем*. Данный метод позволяет справиться с разреженностью данных, возникающих в языках с богатой морфологией: например, доля несловарных слов в корпусе объемом $65 \cdot 10^3$ токенов для английского языка составляет 1.2%, тогда как для русского этот показатель намного больше — 7.5%. Еще хуже ситуация обстоит с финским, турецким и арабским [19].

Выделение подслов может производиться как автоматически [45], так и на основе экспертных знаний.

В [46] приводятся довольно противоречивые результаты использования данного подхода. Разбиение на подслова производилось автоматически без учителя. Авторы утверждают, что по сравнению с использованием триграммной модели на токенах, модель, использующая подслова, обладает меньшей долей несловарных слов в корпусе (0.02% на подсловах против 20% на токенах) и дает сильно меньший уровень пословной ошибки (31% против 56%). Претензии к авторам могут быть высказаны по двум пунктам: во-первых, доля несловарных слов не является метрикой качества для системы распознавания речи, и ее снижение не может считаться признаком успешности модели; во-вторых исходный 56%-й WER свидетельствует о том, что изначально система практически не работала, и *baseline* явно не является адекватным (31% вообще говоря, свидетельствует о том, что она не работала и после).

[19] свидетельствует о том, что модель с использованием подслов, определяемых словарно на основе грамматической теории, также не дают существенного снижения энтропии: 0.12 бит для русского языка.

Еще одним важным применением морфологического подхода к статистическому моделированию языка является использование лемматизации. Выше уже отмечалось, что большое

разнообразие форм слов в русском языке закономерно ведет к большей разреженности данных. В некоторой степени данная проблема может быть упрощена путем приведения всех слов к канонической словарной форме — лемме. Безусловно, неверно было бы ожидать хорошего качества от языковой модели, натренированной исключительно на леммах, однако модель может оказаться полезной при интерполяции.

Как и в случае с определением частей речи, лемматизация является сравнительно дорогой операцией, осуществляемой отдельным алгоритмом. В настоящее время методы морфологического анализа в основном работают аналогично скрытым марковским моделям. Помимо собственно скрытых марковских моделей используются также марковские модели максимальной энтропии [47] и условные случайные поля [48]. При этом использование моделей на леммах и как самостоятельной модели, и при ее интерполяции не дало положительного эффекта для чешского языка, понизив WER лишь на 0.21% [45] с 30%.

1.5.2 Модели, использующие синтаксическую информацию

Использование синтаксического дерева является логичным ответом на проблему ограниченности n-граммы. Как уже отмечалось выше, данная проблема кажется более острой для флективных языков, к которым относится русский.

Рассмотрим синтаксический разбор в виде дерева зависимостей [41] предложения *Глава должна предусматривать этот пункт.*

Суть синтаксического разбора в форме дерева зависимостей состоит в том, что для каждой пары токенов, в случае если между ними имеется синтаксическая связь, когда один токен (главный) «приписывает» другому токену (зависимому) грамматическую форму, то от главного токена к зависимому проводится ориентированное ребро. Таким образом, дерево зависимостей является ориентированным деревом, в каждом узле которого находится токен. Приписав каждому ребру вероятностный вес, т.е. вероятность $\mathsf{P}(CHILD = w_i | PARENT = w_k)$ того, что имеется синтаксическая зависимость между w_t и w_k мы приходим к тому, что вероятность правильного синтаксического разбора можно записать так:

$$\mathsf{P}(w_1, \ldots, w_t, T) = \mathsf{P}(TOP = w_k) \prod_{1<i,j<t} \mathsf{P}^{\delta_T(i,j)}(CHILD = w_i | PARENT = w_j)$$

,

где $\delta_T a(i,j) = 1$ тогда и только тогда, когда w_j является родителем w_i в синтаксическом дереве T. Токен w_k является вершиной в дереве T.

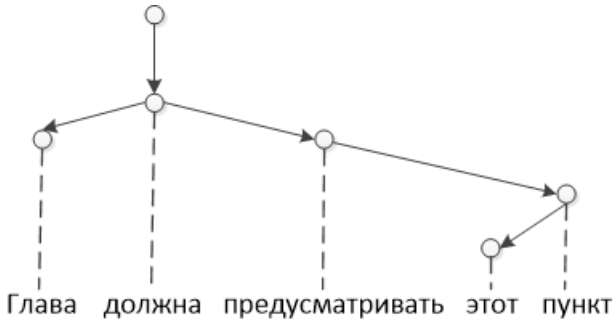

Рисунок 1.3: Синтаксический разбор предложения в структуре зависимостей

Тогда для дерева на рис. 3.1

$$\begin{aligned}\mathsf{P}(w_1 \dots w_t, T) = \mathsf{P}(TOP = \textit{должна})\,\mathsf{P}(CHILD = \textit{предусматривать} | PARENT = \textit{должна}) \times \\ \times\, \mathsf{P}(CHILD = \textit{пункт} | PARENT = \textit{предусматривать}) \times \\ \times\, \mathsf{P}(CHILD = \textit{этот} | PARENT = \textit{пункт}) \times \\ \times\, \mathsf{P}(CHILD = \textit{глава} | PARENT = \textit{предусматривать})\end{aligned}$$

Данный способ факторизации вероятности разбора не единственный [49]. Алгоритмы, основанные на применении условных случайных полей используют в качестве весов ребер взвешенные признаки пар токенов, как в модели максимальной энтропии. [50].

Очевидно, что для правильно обученного синтаксического анализатора разборы вариантов распознавания *Глава должна предусматривать этот пункт* и *Глава должно предусматривать этот пункт* не будут равновероятны. Наилучший вариант синтаксического разбора первого предложения должен иметь большую вероятность, чем наилучший разбор второго. Использование данного факта может помочь в ранжировании гипотез.

В обзоре [51] указано, что достоверные свидетельства удачного применения синтаксической модели, отсутствуют. С другой стороны, наилучшая модель из [52] позволила понизить перплексию с 356 для триграмной модели до 244 — для интеполяции триграмной и синтаксической моделей, что соответствует снижению на 0.54 бита. Столь же впечатляющими являются данные для турецкого языка — 0.65 бита. Однако, объем тренировочных и тестовых данных лишь порядка 10^5 токенов.

Таким образом, можно предположить, что как и в случае с морфологической моделью, синтаксическая модель способна дать выигрыш на небольших выборках.

1.5.3 Факторная модель

Наиболее сложной моделью, учитывающей лингвистические факторы, является факторная модель и ее модификации [53]. Изначально факторная модель была предложена для арабского языка [54].

По сути факторная модель представляет собой развитие и обобщение идеи, заложенной в модели с классами и соответственно подходов к сглаживанию с отступом, которое должно производиться с учетом большого числа предикторных переменных. В своей аргументации авторы апеллируют к теории графических вероятностных моделей, одной из важных задач которой является применение аппарата теории графов к построению и вычислениям сложных вероятностных распределений. [55] В рамках терминологии, принятой в теории

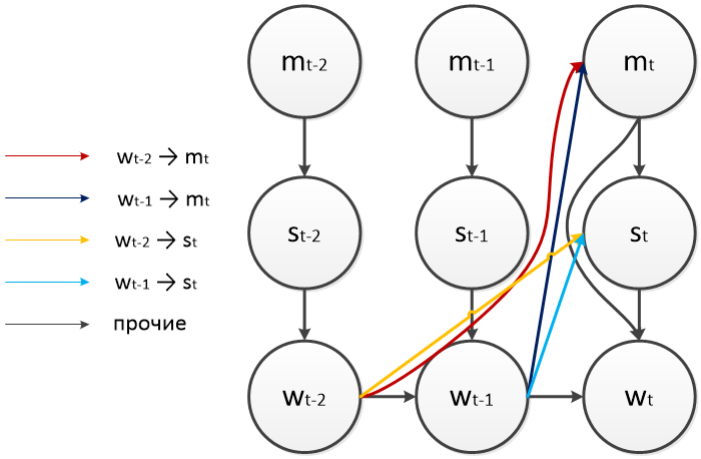

Рисунок 1.4: Графическая модель условного распределения w_t.

графических вероятностных моделей, факторная модель является *динамической байесовской сетью*. Байесовские сети относятся к *ориентированным графическим моделям*. Это означает, что вероятность $\mathsf{P}(w_t|w_1,\ldots,w_{t-1})$ факторизуется в произведение условных и априорных вероятностей (в противоположность неориентированной модели, где известны лишь условные вероятности). Схематично это изображено на рисунке. Фактически же это приводит к формуле, похожей на применяемую в моделях с классами:

$$\hat{\mathsf{P}}(w_t|w_1,\ldots,w_{t-1}) = \mathsf{P}(w_t|s_t,m_t)\,\mathsf{P}(s_t|m_t,w_{t-1},w_{t-2})\,\mathsf{P}(m_t|w_{t-1},w_{t-2}) \tag{1.15}$$

И, как и в случае с классами, мы можем *отступить* к более слабой модели, в случае, если не собрали достаточной статистики для оценки какого либо из сомножителей в произведении (1.15). Основной проблемой, которую пытаются решить авторы модели является выбор оптимальной стратегии для отступа. В [53] предлагается так называемый *обобщенный алгоритм отсутупления (generalized backoff)*. Дело в том, что из схемы, приведенной на рис. 1.4 видно, что отступ можно делать несколькими путями. Поэтому формулы Катца и Кнесера-Нея рассматриваюстя в обощенном виде:

$$P_{GBO}(w_t|w_{t-n+1},\ldots,w_t) = \begin{cases} D_{f,f_1,\ldots,f_K}\frac{C(f,f_1,\ldots,f_K)}{(f_1,\ldots,f_K)}, & \text{если } C(f,f_1,\ldots,f_n) > \tau; \\ \alpha(f_1,\ldots,f_K)\cdot g(f,f_1,\ldots,f_K), & \text{в противном случае} \end{cases} \tag{1.16}$$

В данной формуле, токены $w_1,\ldots,w_t$ заменены на *признаки* $f_1,\ldots,f_K$ — все предикторные переменные, участвующие в предсказании w_t, включая токены и их морфологические, семан-

тические и частотные классы. По большому счету в виде (1.16) формула не говорит ничего нового относительно стандартной модели с отступом [10]. Формула наполняется смыслом при выборе функции $g(f_1 \dots f_K)$, которая и управляет выбором пути в графе на рис. 1.4. В [53] предлагается большое количество эвристических вариантов выбора g. Среди них выбор по максимальной частоте в обучающей выборке: т.е. выбирается модель с самой достоверной оценкой вероятности и ее варианты с различной нормировкой.

Результаты факторной модели остаются неясными. Так, в эксперименте с арабским языком [56] было получено лишь 5%-е снижение перплексии относительно триграммной модели. Эксперименты с английским языком не проводились, однако перспективы модели ориентированные на морфологию с языком, где морфология почти отсутствует, представляются сомнительными.

По большому счету, факторная модель не предлагает ничего нового относительно модели с классами. Основным отличием является ориентированность на терминологию графических вероятностных моделей и соответствующим сведением задачи моделирования языка к поиску в графе. Использование байесовской сети вызывает вопросы ввиду возможного наличия корреляций между признаками (узлами-родителями в терминах графических вероятностных моделей) при их значительном количестве и сложности. Наличие таких корреляций — известный аргумент против использования наивного байсовского классификатора.

1.6 Модели на искусственных нейронных сетях

В 2010 году [57] модели языка, использующие различные архитектуры нейронных сетей, достигли прорывных результатов, продемонстрировав в эксперименте лучшие показатели перплексии, чем 5-граммная модель Кнесера-Нея с кэшированием на различных наборах данных. На больших обучающих корпусах исследователю из университета Брно Т.Миколову удалось добиться значительного снижения не только перплексии, но и уровня пословной ошибки, причем в качестве *baseline* использовалась *state-of-the-art* система распознавания компании IBM.

Нейронные сети использовались в моделировании языка задолго до работ Т.Миколова. Так, в уже неоднократно упоминавшемсся исследовании [11] авторы выражают осторожный оптимизм по поводу нейронных сетей и признаются, что не имели возможности поставить серьезные эксперименты с данной моделью. Нейронные сети по-настоящему привлекли внимание исследователей моделирования языка в 2001 году после выхода статьи И.Бенджио [58], продемонстрировавшей, что нейронная сеть принимающая на вход вектор токенов в окне длины 6 дает лучший результат по перплексии, чем 5-граммная модель со сглаживанием Кнесера-Нея при меньшем размере модели.

В работе [58] впервые была продемонстрирована ставшая впоследствии стандартной методика, согласно которой каждому слову из словаря V ставился в соответствие некоторый вектор $\mathbb{R}^n$. Далее к каждому из векторов применялось нелинейное преобразование f, после

чего на выходном слое на основе линейной комбинации векторов вычислялись вероятности слов из словаря.

$$\mathsf{P}(w_t|w_{t-1},\ldots,w_{t-n+1}) = \frac{e^{y_{w_t}}}{\sum_i e^{y_i}},$$

где

$$y = b + W \cdot x + U \cdot \tanh(d + H \cdot x)$$

— активации нейронной сети на выходном слое. U — матрица весов скрытого слоя; H — матрица весов первого слоя, b,d — векторы смещения, x — векторное представление левого контекста, полученное конкатенацией $n-1$ отображений c_k входных слов в пространство меньшей размерности, полученных согласно формуле:

$$c_k = C^T \delta_i,$$

где δ_{w_i} — единичный вектор с i-й единичной координатой, соответствующий i-му слову в словаре. $C_{|V|\times m}$ — матрица отображения $\mathcal{C}$.

В действительности более ранние эксперименты по использованию нейронных сетей для предсказания слов в тексте относятся к 1990 году. В работе [59] в рамках разработки нейросетевой модели, способной предсказывать произвольные последовательности, автор модели рекуррентной нейронной сети Элман проводил испытания в том числе и на текстах на естественном языке.

Отличием модели Элмана являлось наличие *рекуррентного* слоя: вычисленный первый слой h_t на t-м шаге поступал на вход сети на $t+1$-м шаге. Таким образом, сеть могла сохранять внутреннее состояние.

Именно модель Элмана с незначительными модификациями была через 20 лет успешно использована Т.Миколовым. Модель Т.Миколова оказалась успешной также и для построения языковой модели чешского языка, причем полученные результаты по перплексии оказались лучше [13], чем у лесов решающих деревьев в [56].

1.7 Выводы

На основании анализа, приведенного выше, можно сделать следующие выводы.

1. Основными метриками качества в оценке языковых моделей являются перплексия и уровень пословной ошибки (для распознавания речи). Ни один из этих показателей не является исчерпывающим: недостатком перплексии является предположение о полной и истинной информации о левом контексте, что не соответствует действительности для задачи распознавания речи. Снижение уровня пословной ошибки, в свою очередь, зависит от изначальной конфигурации системы распознавания, что делает результаты, полученные разными исследовательскими группами, несравнимыми.

2. Данные по перплексии не являются самодостаточными, поскольку для разных значений перплексии снижение энтропии может значительно различаться. При этом есть основания считать, что снижение уровня ошибки линейно зависит именно от энтропии.

3. Сглаженная n-граммная модель работает достаточно хорошо, причем с ростом объема обучающей выборки перплексия, как и уровень пословной ошибки, падают.

4. Применение модели, допускающей перестановки, не является оправданным даже для флективных языков, так как свободный порядок слов подчиняется достаточно жестким ограничениям.

5. Среди моделей, не использующих векторное представление словаря, наиболее эффективными являются модели с кэшированием, однако применение кэширования для распознавания речи затруднено ввиду проблемы «закрепления ошибки», когда неверное распознавания одного слова, существенно влияет на качество распознавания последующих слов. То же верно для модели максимальной энтропии в случае. когда используются соответствующие признаки.

6. Более сложные техники оказываются полезны прежде всего для борьбы с разреженностью данных. С ростом объема обучающей выборки для большинства техник наблюдается постепенное уменьшение разности в перплексии сглаженных n-граммных моделей и их интерполяции с моделями, использующими дополнительную информацию.

7. Факторная модель по сути представляет собой унифицированный способ использования различной статистической информации в рамках общей генеративной модели. В этом смысле факторная модель наследует как преимущества, так и недостатки всех моделей, которые она включает в себя.

8. Эффективными оказываются техники, основанные на распределенном представлении словаря, т.е. те техники, в которых каждая словоформа отображается в n-мерный вектор. К таким техникам относятся языковые модели, основанные на латентном семантическом анализе, латентном размещении Дирихле, нейронных сетях и тематических моделях.

9. Использование специальных лингвистических данных оправданно для языков с богатой морфологией и при сравнительно небольших объемах обучающих данных, когда невозможно обеспечить корректное обучение для словаря объемом почти в 10 раз большего, чем в случае английского языка.

10. Среди алгоритмов классификации наибольшую эффективность показывают леса случайных деревьев.

11. Наилучшие результаты были достигнуты при помощи моделей, использующих рекуррентные нейронные сети. Важнейшим преимуществом таких моделей является распределенное представление словаря, однако обучение нейронных сетей представляет собой отдельную достаточно сложную задачу.

12. Комбинации наиболее эффективных техник дают результаты, как минимум не хуже, чем каждая и з техник в отдельности. Эффективный способ комбинирования является важной отдельной задачей.

13. В отсутствии больших обучающих корпусов представляется логичным комбинирование некоторой сложной модели, основанной на применении лингвистической информации и векторных представлениях, полученных применением тематических моделей и нейронных сетей.

Глава 2

Моделирование языка с помощью рекуррентных нейронных сетей

2.1 Введение

В предыдущей главе был сделан вывод о том, что наиболее перспективными методами моделирования языка на сегодняшний день являются методы, основанные на рекуррентных нейронных сетях. В данной главе эти методы будут рассмотрены подробно. Глава структурирована следующим образом. В разделе 2 будет описан общий подход к моделированию языка при помощи нейронных сетей. В разделе 3 будет подробно описана рекуррентная нейронная сеть. Раздел 4 посвящен ограничениям рекуррентных нейронных сетей, связанных с проблемой затухания (*vanishing*) и взрыва (*explosion*) градиента. В разделе 5 описаны различные подходы к решению данных проблем. Части 6 и 7 посвящены применимости рекуррентных нейронных сетей к моделированию флективных языков. Наконец, в разделе 8 делаются основные выводы.

2.2 Общие принципы языкового моделирования при помощи нейронных сетей

Как отмечалось в предыдущей главе, первым известным примером удачного применения нейронных сетей к моделированию языка является работа И.Бенджио 2003 года [58]. Именно в ней был изложен подход к получению векторных представлений слов, ставший впоследствии общепринятым. Рассмотрим его подробно.

Пусть $\mathbb{V}$ — множество слов в словаре моделируемого языка $\mathcal{L}$. В качестве целевой функции нейронной сети f возьмем $\mathsf{P}(w_t|w_{t-n+1}^{t-1})$. Далее разложим $f(w_t,w_{t-1},\ldots,w_{t-n+1})$ на две составляющие:

1. Отображение $\mathcal{C} : \mathbb{V} \to \mathbb{R}^m$. $C(w_i)$ является векторным представлением слова w_i. C определяется матрицей размерности $|\mathbb{V}| \times m$.

2. Функция $g : (C(w_{t-n+1}),\ldots,C(w_{t-1})) \to Y$, где $Y = \{y_1 \ldots y_{|\mathbb{V}|} | \sum_{i=1}^{|\mathbb{V}|} y_i = 1\}$. Y можно интерпретировать как распределение вероятностей: $y_i = p(w_t = v_i | w_{t-n+1} \ldots w_{t-1}) \forall v_i \in \mathbb{V}$.

В качестве функции ошибки возьмем регуляризованную кросс-энтропию:

$$L(\theta) = \frac{1}{T} \sum_t \log f(w_t,w_{t-1},\ldots,w_{t-n+1};\theta) + R(\theta),$$

где $R(\theta)$ — регуляризационный член.

Поскольку выбранная функция f должна удовлетворять критерию $\sum_v f(v,w_{t-1},\ldots,w_{t-n+1}) = 1$ удобно использовать софтмакс-активацию на выходном слое нейронной сети:

$$f(v,w_{t-1},\ldots,w_{t-n+1}) = \frac{e^{y_v}}{\sum_{v'} e^{y_{v'}}}. \tag{2.1}$$

Значения y_i на выходном слое вычисляются по формуле:

$$y = b + D \cdot x + V \cdot \tanh(d + H \cdot x), \tag{2.2}$$

где V — матрица весов выходного слоя; H — матрица весов скрытого слоя, b,d — векторы смещения, x — входной вектор, D — матрица «прямых соединений» (*direct connections*), возможно состоящая только из нулей.

Вектор x представляет собой конкатенацию векторных представлений $C(w_i)$ $n-1$ слов левого контекста.

$$C(w_i) = C^T \delta_{w_i},$$

где δ_{w_i} — единичный вектор с i-й единичной координатой, соответствующий i-му слову в словаре. $C_{|\mathbb{V}| \times m}$ — матрица отображения $\mathcal{C}$.

Таким образом, основной идеей языкового моделирования при помощи нейронных сетей является разбиение функции распределения на 2 компоненты: 1) функцию отображения $\mathcal{C} : \mathbb{V} \to \mathbb{R}^m$, задающую соответствие между словом в словаре и его векторным представлением; 2) функцию $g : (C(w_{t-n+1}),\ldots,C(w_{t-1})) \to Y$, где $Y = \{y_1 \ldots y_{|\mathbb{V}|} | \sum_{i=1}^{|\mathbb{V}|} y_i = 1\}$, вычисляющую условное распределение вероятностей слов в словаре на основе векторных представлений слов в левом контексте. На текущий момент данная схема является стандартной для языкового моделирования на нейронных сетях. Она же используется и при выборе рекуррентной архитектуры.

2.3 Рекуррентные нейронные сети

В данном разделе будут рассмотрена классическая архитектура рекуррентной нейронной сети безотносительно к ее применению для языкового моделирования.

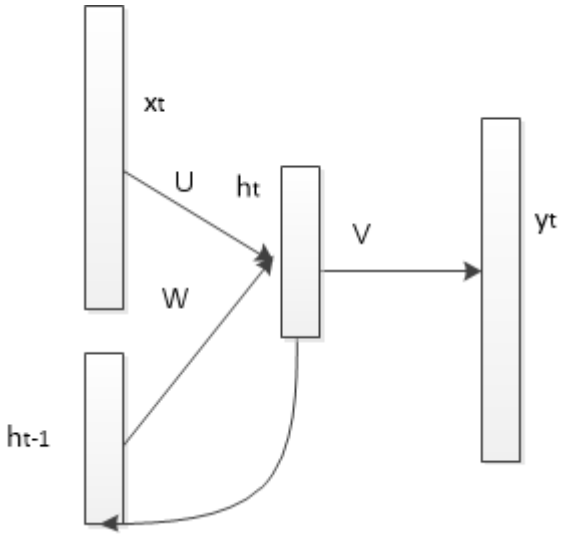

Рисунок 2.1: Общий вид рекурренной сети Элмана. x_t — входной слой на шаге t; h_t — скрытый слой; y_t — выходной слой.

Прежде всего, стоит отметить, что выше под термином *рекуррентная нейронная сеть* далее подразумевается рекуррентная архитектура, предложенная Элманом в 1990 году [59]. Строго говоря, данная архитектура не является единственной. Ее более точное название *сеть Элмана* в противоположность более ранней *сети Джордана* [60] и *сети Хопфилда* [61]. Ниже, однако, для удобства под рекуррентной нейронной сетью будет пониматься исключительно архитектура Элмана.

Рекуррентная сеть Элмана представляет собой двухслойную нейронную сеть, в которой скрытый слой h_t, полученный на шаге t поступает на вход сети на следующем $t+1$ шаге. См. рис. 2.1

Рассмотрим выборку $\mathcal{D}$, состоящую из пар $(x^{(t)}, y^{(t)})$, зависимых временных рядов одинаковой длины. Пусть $(x^{(t)})$ и $(y^{(t)})$ определены соответственно на множествах $\mathbb{U}$ и $\mathbb{T}$. Тогда рекуррентная нейронная сеть есть функция, аппроксимирующая условное распределение $\mathsf{P}(y^{(t)}|x^{(t)})$ согласно формулам:

$$h_t = f(W \cdot h_{t-1} + U \cdot x_t + b) \tag{2.3}$$

$$y_t = g(V \cdot h_t + d) \tag{2.4}$$

W,U,V — матрицы весов, b, d — смещения, $x \in \mathbb{U}$ — элемент предикторной последовательности на шаге t, $y \in \mathbb{T}$ — распределение вероятностей элементов неизвестной последовательности на том же шаге t, $h \in \mathbb{H}$ — скрытый слой сети, f и g — функции активации.

В качестве f часто берется сигмоидная функция, либо выпрямитель (rectifier linear unit, ReLU) [62]. В качестве g — многоклассовая логит-функция (softmax) линейная активация.

Стоит отметить, что хотя формально последовательности $(x^{(t)})$, $(y^{(t)})$ и имеют одинаковую длину, на этапе обучения достаточно вычислять функцию потери только в интересующих нас

точках, следовательно $(x^{(t)})$ и $(y^{(t)})$ фактически могут иметь различную длину. Это становится удобным, например, в задаче оценки эмоциональной тональности текста, когда ошибка может быть вычислена лишь по прочтении всего текста целиком. [63].

2.4 Обучение рекуррентных нейронных сетей

2.4.1 Алгоритм распространения ошибки обратно по времени (BPTT)

На сегодняшний день, основным методом оптимизации, используемым для подбора параметров модели θ нейронной сети, является метод градиентного спуска. В случае подбора параметров нейронной сети градиентный спуск приводит к известному алгоритму обратного распространения ошибки [8].

Аналогичный подход для рекуррентных нейронных сетей приводит к т.н. *алгоритму распространения ошибки обратно по времени* (*Backpropagation through time*, BPTT) [64].

Рассмотрим выход рекуррентной нейронной сети на некотором шаге t:

$$\begin{aligned} y_t = g(V \cdot h_t + d) = g(V \cdot f(U \cdot x_t + W \cdot h_{t-1} + b) + d) = \\ g(V \cdot f(U \cdot x_t + W \cdot f(U \cdot x_{t-1} + W \cdot h_{t-2} + b) + b) + d) = \\ g(V \cdot f(U \cdot x_t + W \cdot f(U \cdot x_{t-1} + W \cdot f(...f(U \cdot x_1 + W \cdot h_0)...) + b) + b) + d) \end{aligned} \tag{2.5}$$

Определим функцию ошибки $\mathcal{L}$ как сумму ошибок на каждом шаге пары последовательностей:

$$\mathcal{L}(\theta) = \sum_{1 \le t \le T} \mathcal{L}_t(\theta) \tag{2.6}$$

Получим выражения для градиентов $\frac{\partial \mathcal{L}}{\partial \theta}$:

$$\frac{\partial \mathcal{L}}{\partial V} = \sum_{1 \le t \le T} \frac{\partial \mathcal{L}_t}{\partial y_t} \frac{\partial g(V \cdot h_t + d)}{\partial V} \tag{2.7}$$

$$\frac{\partial \mathcal{L}}{\partial d} = \sum_{1 \le t \le T} \frac{\partial \mathcal{L}_t}{\partial y_t} \frac{\partial g(V \cdot h_t + d)}{\partial d} \tag{2.8}$$

Из 2.5 видно, что выражение для $\frac{\partial \mathcal{L}}{\partial W}$ может быть получено в явном виде, если 2.5 расписать полностью от 1 до t. Если вернуться к представлению 2.5 в виде нейронной сети, то мы получим граф, изображенный на рис.2.2. Фактически мы представили вычисление выхода y_t как результат полного цикла работы многослойной нейронной сети с $t-1$ слоями и одинаковой матрицей синапсов W. Таким образом, для вычисления $\frac{\partial \mathcal{L}}{\partial W}$ мы будем использовать алгоритм обратного распространения ошибки на нейронной сети, полученной путем развертки исходной рекуррентной сети по времени.

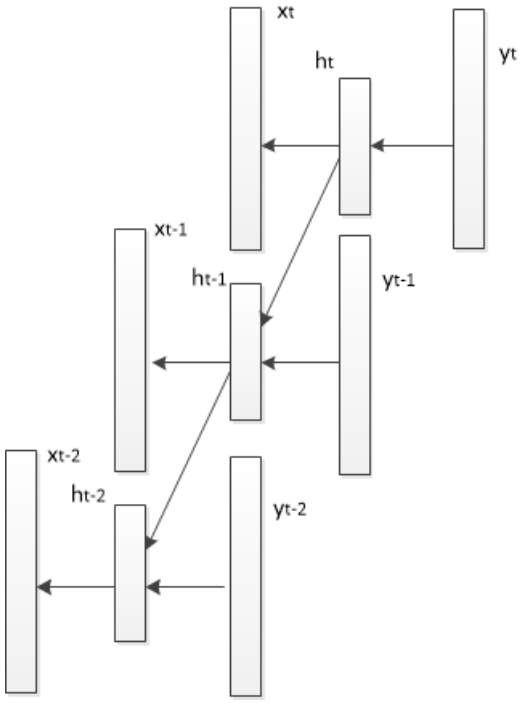

Рисунок 2.2: Схема вычислений в алгоритме распространения ошибки обратно по времени. y_k — выходной слой на шаге t; x_t входной слой; h_t — скрытый слой. Стрелка указывает направление распространения ошибки. Развертка сети по времени производится на 2 шага.

Чтобы перейти к более формальным рассуждениям для удобства введем понятие *мгновенной производной.*

Определение 1 (Мгновенная производная)**.** *Пусть h_t вычислено рекурсивно согласно уравнению $h_t = f(h_{t-1},\theta)$. Мгновенной частной производной $\frac{\partial^+ h_t}{\partial \theta}$ называется производная h_t по θ, если принимается, что $\frac{\partial h_{t-1}}{\partial \theta} = 0$.*

С учетом опр. 1 градиент $\frac{\partial \mathcal{L}}{\partial W}$ можно записать как:

$$\frac{\partial \mathcal{L}}{\partial W} = \sum_{t=1}^{T} \frac{\partial \mathcal{L}_t}{\partial h_t}\frac{\partial h_t}{\partial W} =$$

$$= \frac{\partial \mathcal{L}_T}{\partial h_T}\frac{\partial^+ h_T}{\partial W} + \frac{\partial \mathcal{L}_T}{\partial h_T}\frac{\partial h_T}{\partial h_{T-1}}\frac{\partial^+ h_{T-1}}{\partial W} + \sum_{t=1}^{T-2} \frac{\partial \mathcal{L}_T}{\partial h_T}\frac{\partial h_T}{\partial h_t}\frac{\partial^+ h_t}{\partial W} +$$

$$+ \frac{\partial \mathcal{L}_{T-1}}{\partial h_{T-1}}\frac{\partial^+ h_{T-1}}{\partial W} + \sum_{t=1}^{T-2} \frac{\partial \mathcal{L}_{T-1}}{\partial h_{T-1}}\frac{\partial h_{T-1}}{\partial h_t}\frac{\partial^+ h_t}{\partial W} + \cdots + \frac{\partial \mathcal{L}_t}{\partial h_t}\frac{\partial h_1}{\partial W} =$$

$$= \sum_{t=1}^{T} \frac{\partial \mathcal{L}_t}{\partial h_t} \sum_{k=1}^{t} \frac{\partial h_t}{\partial h_k}\frac{\partial^+ h_k}{\partial W} \quad (2.9)$$

Заметим также, что из определения мгновенной производной и 2.9 следует, что

$$\frac{\partial^+ h_{T-1}}{\partial W} = \frac{\partial^+ h_T}{\partial W}\frac{\partial h_T}{\partial h_{T-1}} + \frac{\partial \mathcal{L}_{T-1}}{\partial h_{T-1}} \quad (2.10)$$

Используя формулы выше, можно предложить следующий эффективный алгоритм обучения рекуррентной нейронной сети [65]. См. алгоритм 2.4.1.

Algorithm 2.4.1 Алгоритм распространения ошибки обратно по времени (BPTT)

InitRandomMatrices() – функция случайной инициализации матриц модели
InitZeroMatrices() – функция инициализации матриц нулевыми элементами
InitialState() – функция инициализации скрытого состояния (0 или 0.1)

Вход: $\{\mathbf{u}\}, \{\mathbf{t}\}$ // *последовательности* $\boldsymbol{u}$ *входов и меток* $\boldsymbol{t}$

$\mathbf{U}, \mathbf{W}, \mathbf{V} \leftarrow InitRandomMatrices()$
$\mathbf{gW}, \mathbf{gV}, \mathbf{gU} \leftarrow InitZeroMatrices()$
$\mathbf{h}_0, \leftarrow InitialState()$ // *инициализация*
повторять
  для $t = 1, \ldots, N$
    $\mathbf{h}_t \leftarrow f(\mathbf{W} \cdot \mathbf{h}_{t-1} + \mathbf{U} \cdot \mathbf{u}_t)$ // *прямой ход*
    $\mathbf{h}_t \leftarrow g(\mathbf{V} \cdot \mathbf{h}_t)$
  $\mathbf{gV} \leftarrow \partial \mathcal{L}(\mathbf{y}_T, \mathbf{t}_T)/\partial \mathbf{V}$ // *обратный ход*
  $g_{\mathbf{h}} \leftarrow \partial \mathcal{L}(\mathbf{y}_T, \mathbf{t}_T)/\partial \mathbf{h}_T$
  $\mathbf{gU} \leftarrow g_{\mathbf{h}} \cdot (\partial^+ \mathbf{h}_T/\partial \mathbf{U})$
  $\mathbf{gW} \leftarrow g_{\mathbf{h}} \cdot (\partial^+ \mathbf{h}_T/\partial \mathbf{W})$
  для $T-1, T-2, \ldots, 1$
    $g_{\mathbf{h}} \leftarrow g_{\mathbf{h}} \cdot (\partial \mathbf{h}_{t+1}/\partial \mathbf{h}_t) + \partial \mathcal{L}(\mathbf{y}_t, \mathbf{t}_t)/\partial \mathbf{h}_t$
    $\mathbf{gV} \leftarrow \mathbf{gV} + (\partial \mathcal{L}(\mathbf{y}_t, \mathbf{t}_t)/\partial \mathbf{V})$
    $\mathbf{gU} \leftarrow \mathbf{gU} + g_{\mathbf{h}} \cdot (\partial^+ \mathbf{h}_t/\partial \mathbf{U})$
    $\mathbf{gW} \leftarrow \mathbf{gW} + g_{\mathbf{h}} \cdot (\partial^+ \mathbf{h}_t/\partial \mathbf{W})$
  $U \leftarrow U + \alpha \mathbf{gU}$ // *обновление модели*
  $U \leftarrow V + \alpha \mathbf{gV}$
  $U \leftarrow W + \alpha \mathbf{gW}$
пока Не выполнено условие остановки

Данный алгоритм обладает сложностью $\Theta(T)$ по длине входной последовательности [65]. Ниже приведены формулы для вычисления градиентов на каждом шаге развертки сети.

$$\mathbf{e}_y = \frac{\partial \mathcal{L}_t}{\partial y_t} g\prime(V \cdot h_t) \tag{2.11}$$

$$\frac{\partial \mathcal{L}}{\partial V} = \mathbf{e}_y \cdot \mathbf{h}_t^T \tag{2.12}$$

$$\mathbf{e}_h[t] = (V^T \cdot \mathbf{e}_y + W^T \cdot \mathbf{e}_h[t+1]) \cdot diag(f'(h_{t-1})) \tag{2.13}$$

$$\frac{\partial \mathcal{L}}{\partial W} = \mathbf{e}_h[t] \cdot \mathbf{h}_{t-1}^T \tag{2.14}$$

$$\frac{\partial \mathcal{L}}{\partial U} = \mathbf{e}_h[t] \cdot \mathbf{u}_t^T \tag{2.15}$$

Как показано в алгоритме 2.4.1 градиенты суммируются на каждом шаге.

2.4.2 Ограничения алгоритма распространения ошибки обратно по времени

Более подробный анализ формулы 2.9 позволяет выявить важную фундаментальную проблему алгоритма распространения ошибки обратно по времени. А именно, тот факт, что данный алгоритм не способен учитывать влияние предшествующих элементов последовательности, если они находятся достаточно далеко от текущего. Действительно, подстановкой 2.3 в 2.9 получим:

$$\frac{\partial h_t}{\partial h_k} = \prod_{k<i\leq t} W^T \cdot diag(f'(h_{i-1})) \tag{2.16}$$

Ниже будет показано, что в зависимости от свойств матрицы W значение выражения 2.16 либо растет, либо падает с экспоненциальной скоростью. Данный факт получил название затухания градиента (*vanishing gradient*) в случае убывания или градиентного взрыва (*gradient explosion*) в случае роста. Рассмотрим более важный для дальнейшего изложения случай затухания. Докажем следующее утверждение:

Теорема 1 (О затухании градиента)**.** *Существует такая инициализация рекуррентной нейронной сети со скрытым слоем*

$$h_t = \sigma((W \cdot h_{t-1} + U \cdot x_t + b) \tag{2.17}$$

и сигмоидной функцией активации σ, что норма градиента $\frac{\partial h_t}{\partial h_k}$ для $k < t$ экспоненциально стремится к нулю с ростом $t - k$.

Доказательство. Без ограничения общности уравнение 2.17 можно переписать в виде:

$$h'_t = W \cdot \sigma(h'_{t-1}) + U \cdot x_t + b$$

— далее в доказательстве в качестве h_t используется h'_t из уравнения выше. Рассмотрим градиент 2.16 с сигмоидной функцией активации $\sigma(h) = \frac{1}{1+e^{-h}}$, применяемой к вектору h поэлементно:

$$\frac{\partial h_t}{\partial h_k} = \prod_{k<i\leq t} W^T \cdot diag(\sigma'(h_{i-1})) = \prod_{k<i\leq t} W^T \cdot diag(\sigma(h_{i-1})(1 - \sigma(h_{i-1})))$$

Поскольку $\sigma'(x) = \sigma(x)(1 - \sigma(x)) \leq 1/4 = c$,

$$\frac{\partial h_t}{\partial h_k} = \prod_{k<i\leq t} W^T \cdot diag(\sigma'(h_{i-1})) \leq (cW^T)^{t-k}$$

В общем случае W^T не диагонализуема, поэтому рассмотрим жорданову форму W^T в $\mathbb{C}$:

$$W^T = S^{-1}AS$$

$$(cW^T)^{t-k} = S^{-1}(cA)^{t-k}S$$

Пусть

$$A = \begin{pmatrix} \lambda_1 & 1 & & \\ & \lambda_1 & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_k \end{pmatrix}$$

жорданова матрица с собственными значениями $\lambda_1 \dots \lambda_k$. Рассмотрим жорданову клетку $\mathbf{A_0}_{r_0 \times r_0}$ с собственным значением $\lambda_0 > \lambda_i \forall i$.

$$c^p {\mathbf{A_0}}^p = c^p \begin{pmatrix} \lambda_0^p & \binom{p}{1}\lambda_0^{p-1} & \binom{p}{2}\lambda_0^{p-2} & \dots \\ & \lambda_0^p & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_0^p \end{pmatrix} = \begin{pmatrix} (c\lambda_0)^p & c\binom{p}{1}(c\lambda_0)^{p-1} & c^2\binom{p}{2}(c\lambda_0)^{p-2} & \dots \\ & (c\lambda_0)^p & \ddots & \\ & & \ddots & c\binom{p}{l}(c\lambda_0)^{p-l} \\ & & & (c\lambda_0)^p \end{pmatrix} =$$

$$= \begin{pmatrix} q^p & c\binom{p}{1}q^{p-1} & c^2\binom{p}{2}q^{p-2} & \dots \\ & q^p & \ddots & \\ & & \ddots & c\binom{n}{1}(q)^{p-l} \\ & & & q^p \end{pmatrix}$$

Рассмотрим случай $q < 1$. Для сигмоидной функции активации, это справедливо при $\lambda < 4$. В этом случае норма Фробениуса $\|c^p {\mathbf{A_0}}^p\|_2^2$ ограничена сверху:

$$\|c^p {\mathbf{A_0}}^p\|_2^2 < C \cdot \binom{p}{r_0 - 1}^2 q^{2p}$$

$$C \cdot \binom{p}{r_0 - 1}^2 q^{2p} < C/r_0! \cdot p^{2(r_0)} q^{2p}$$

Поскольку q^p убывает быстрее, чем растет p^{r_0}, произведение $q^p p^{r_0} \to 0$ с ростом p. Значение $(cA)^{t-k}$ ограничено сверху степенью жордановой клетки с наибольшим собственным значением. Таким образом, норма степени $(W^T)^{t-k}$ экспоненциально стремится к нулю с ростом $t - k$. $\square$

Другими словами, поведение градиента $\partial h_t / \partial h_k$ зависит от наибольшего собственного значения матрицы W.

Данный результат был получен независимо в работах [66], [67]. В [66] невыполнение условия (достаточного), приведенного в теореме выше является *необходимым* для экспоненциального роста нормы $\frac{\partial h_t}{\partial h_k}$, приводящего к проблеме градиентного взрыва.

В приведенном доказательстве мы опирались на тот факт, что производная сигмоидной функции ограничена сверху. То же верно для гиперболического тангенса ($f'(x) \leq 1$) и линейного выпрямителя ($f'(x) = 1$).

Фактически данный результат означает, что тренировка рекуррентной нейронной сети методами первого порядка не может учитывать влияния элементов последовательности, если они сильно разнесены по времени. Для этого матрица W должна была бы иметь достаточно большую норму, а значит быть критически восприимчивой к шуму в обучающей последовательности ([66]). На практике это выражается в высокой амплитуде норм градиентов и неустойчивости решения. С другой стороны, устойчивое решение может быть получено при небольших нормах W, однако, как было показано выше, такие решения приводят к сложностям с моделированием дальних зависимостей.

2.5 Подходы к решению проблемы моделирования дальних зависимостей

Начиная с 1994 года, когда в [66] описанное выше ограничение алгоритма BPTT было впервые обоснованно, были предложены различные методы решения проблемы затухания или неустойчивости градиента.

Очевидным способом борьбы с неустойчивостью является инициализация матрицы W достаточно малыми значениями. Также можно использовать L1 и L2-нормы, чтобы предотвратить всплеск нормы градиента в процессе обучения. Как уже отмечалось выше, данное решение закономерно приводит к противоположной проблеме — затуханию градиента.

Другой простейшей техникой является обрезка (*clipping*) градиента по некоторому заранее заданному порогу. Данная техника была предложена в [13] и исследована в [68]. В последней работе утверждается, что несмотря на свою простоту, данное решение является эффективным и имеет теоретическое обоснование.

Для борьбы с затухающими градиентами в [68] было предложено использовать аддитивный регуляризатор:

$$\Omega = \sum_k \Omega_k = \sum_k \left(\frac{\left\| \frac{\partial \mathcal{L}}{\partial h_{k+1}} \frac{\partial h_{k+1}}{\partial h_k} \right\|}{\left\| \frac{\partial \mathcal{L}}{\partial h_{k+1}} \right\|} - 1\right)^2$$

Функция данного регуляризатора состоит в том, чтобы дать преимущество тем направлениям градиентного спуска, на которых норма градиента, вычисленного на предыдущем шаге, не меняется значительно. То есть на каждом k-м шаге выбирается направление, не только минимизирующее ошибку, но и сходное с уже выбранным ранее. Авторы утверждают, что стратегия обучения в данном случае состоит в том, чтобы вначале научить сеть «запоминать» предыдущие элементы — для этого нельзя двигаться исключительно в направлении градиента — и использовать регуляризатор. На следующей итерации нужно научить сеть «забывать» нерелевантные входы. В этом случае затухание градиента как раз желательно, и регуляризатор не используется.

В [69] было предложено отказаться от метода градиентного спуска и использовать методы второго порядка. В частности, Hessian Free. Данный метод позволил получить хорошие

результаты в т.н. *патологических задачах* (pathological problems), правильное выполнение которых невозможно без моделирования дальних зависимостей. К таким задачам относятся, например, *проблема классификации по порядку* (temporal order problem), суть которой состоит в том, чтобы классифицировать последовательности произвольной длины в зависимости от того, в каком порядке в ней появляются 2 ключевых элемента – A и B. Элементы могут стоять в любом месте последовательности. Тем не менее, в связи со сложностью реализации и невысокой скоростью работы, данный метод почти не используется на практике.

В [67] было предложено вовсе отказаться от классической архитектуры рекуррентной нейронной сети и заменить элементы скрытого слоя с нейронов на сложные *ячейки памяти*. Данная архитектура получила название модели с *длительной кратковременной памятью* (*Long short-term memory*, LSTM). На текущий момент LSTM успешно применяется в различных областях, включая генерацию текста [70] и моделирование языка [71].

Авторы перечисленных методов в основном фокусируются на улучшении результатов в патологических задачах. Тем не менее, успешность n-граммных моделей позволяет предположить, что для моделирования языка зависимости произвольной длины не так существенны. В следующем разделе будет рассмотрен стандартный подход к языковому моделированию при помощи рекуррентных нейронных сетей, после чего будет рассмотрен вопрос о серьезности проблемы угасания градиента для моделирования языка.

2.6 Моделирование языка с помощью рекуррентных нейронных сетей

Выше задача, решаемая рекуррентной нейронной сетью Элмана, была сформулирована для временных рядов равной длины $(x^{(t)}, y^{(t)})$. Если в качестве последовательности $(y^{(t)})$ взять последовательность $(x^{(t)})$ с задержкой на один такт, то с помощью модели Элмана можно решать задачу прогнозирования временного ряда $(x^{(t)})$. К этой задаче фактически сводится задача языкового моделирования. При этом принимаются следующие положения:

– Значения $(x^{(t)})$ определены на конечном множестве $\mathbb{V}$ словоформ моделируемого языка $\mathcal{L}$. Таким образом, задача прогнозирования решается на дискретном вероятностном пространстве.

– Искомое распределение вероятностей следующего слова $p(x_t|x_1 \dots x_{t-1})$ вычислимо с помощью рекуррентной сети Элмана.

Языковая модель на рекуррентной нейронной сети может быть также представлена как последовательность операций отображения $\mathcal{C} : \mathbb{V}^* \to \mathbb{R}^n$ и линейной классификации $\mathcal{F} : \mathbb{R}^n \to \mathbb{V}$, где в роли классов выступают элементы словаря $\mathbb{V}$. В качестве преобразования $\mathcal{F}$ можно использовать любой алгоритм классификации на $|\mathbb{V}|$ классов. Более того, преобразование $\mathcal{C}$ также может быть выбрано произвольно, однако использование нейросетевого

подхода обладает рядом технических преимуществ: 1) возможностью моделировать сложные нелинейные отображения и 2) наличием простых и эффективных обучающих алгоритмов.

Подход к моделированию языка на основе рекуррентных нейронных сетей фактически представляет собой адаптацию подхода И.Бенджио, описанного в разделе 2.2 к архитектуре Элмана.

Таким образом, основными особенностями подхода являются следующие:

1. Как и в модели Бенджио, входным вектором является вектор δ_i с единственной единичной координатой i.

2. Как и в модели Элмана, скрытый слой h_{t-1} на шаге $t-1$ поступает на вход сети на следующем шаге t.

3. Выходом сети y_t является распределение вероятностей слов $p(w_{t+1}|h_t, w_t)$, вычисляемое как результат softmax-активации на выходном слое y_t.

В работе [57] в качестве активации скрытого слоя была выбрана сигмоидная функция, смещения принимались равными нулю. Таким образом, выход рекуррентной нейронной сети для моделирования языка вычисляется согласно формулам:

$$h_t = \sigma(W \cdot h_{t-1} + U \cdot x_t) \tag{2.18}$$

$$y_t = softmax(V \cdot h_t) \tag{2.19}$$

где W,U,V — матрицы весов, $x \in \mathbb{U}$ — элемент предикторной последовательности на шаге t, $y \in \mathbb{T}$ — распределение вероятностей элементов неизвестной последовательности на том же шаге t, $h \in \mathbb{H}$ — скрытый слой сети, σ и $softmax$ — сигмоидная и софтмакс- активации соответственно.

Также можно в явном виде выписать уравнения для ошибок на слое и градиентов в алгоритме BPTT на каждом шаге:

$$\mathbf{e}_y = \mathbf{t}_t - y_t \tag{2.20}$$

$$\frac{\partial \mathcal{L}}{\partial V} = \mathbf{e}_y \cdot \mathbf{h}_t^T \tag{2.21}$$

$$H_a = \begin{pmatrix} h[t]_1(1-h[t]_1) & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & h[t]_H(1-h[t]_H) \end{pmatrix} \tag{2.22}$$

$$\mathbf{e}_h[t] = H_a \cdot (V^T \cdot \mathbf{e}_y + W^T \cdot \mathbf{e}_h[t+1]) \tag{2.23}$$

$$\frac{\partial \mathcal{L}}{\partial W} = \mathbf{e}_h[t] \cdot \mathbf{h}_{t-1}^T \tag{2.24}$$

$$\frac{\partial \mathcal{L}}{\partial U} = \mathbf{e}_h[t] \cdot \mathbf{u}_t^T \tag{2.25}$$

Также в реализации метода используется L2 регуляризация, не отраженная в алгоритме 2.4.1.

Реализация Т.Миколова предполагала оригинальную схему обновления модели:

1. Обновление матрицы V весов между нейронами скрытого и выходного слоя производится на **каждом** временном шаге.

2. Обновление словарной таблицы U и рекуррентной матрицы W производится на каждом k-м шаге.

3. Распространение ошибки обратно по времени производится на $p < k$ шагов назад.

Таким образом, при проходе по обучающей последовательности длины T и параметрах $k = T/2$, $p < T/4$, матрица V будет обновлена T раз, а матрицы U и W по два раза, причем результирующие градиенты $\frac{\partial \mathcal{L}}{\partial W}$ и $\frac{\partial \mathcal{L}}{\partial U}$ будут представлять собой суммы градиентов на предыдущих $T/4$ шагах.

Данный подход можно трактовать как метод стохастического градиента. Его преимуществами являются более быстрая и, как показывают эксперименты, стабильная сходимость. Вместе с тем, использование данного подхода вводит два гиперпараметра – частоту k и глубину p развертывания сети во времени. Значения этих параметров непосредственно связаны с обсуждавшейся выше проблемой затухания градиентов: очевидно, что глубокая развертка не имеет смысла при затухании.

Эксперименты, описываемые в [13] демонстрируют наилучшие результаты при глубине развертки 5-6 шагов.

2.6.1 Модель с частотными классами

Стоит упомянуть одну модификацию рекуррентной нейронной сети, описанную в [13]. Данная рекуррентная архитектура отличается от описанной выше в данной главе в том, что в ней задействованы частотные классы — расширение исходной архитектуры, целью которого являлось ускорение обучения на больших корпусах.

В рамках модели с классами условная вероятность следующего слова вычисляется как:

$$p(w_t|h_t,w_{t-1}) = p(c_t|h_{t-1}, w_t)p(w_t|c_t, w_t, h_{t-1})$$

$$p(c_t|h_{t-1}, w_t) = softmax(V_c^T h_t)$$

$$C = \arg\max_{c}(p(c_t|h_{t-1}, w_t))$$

$$p(w_t|c_t, w_t, h_{t-1}) = softmax(V_w \cdot h'_t),$$

где $V_{c|C|\times|H|}$ — матрица весов классов, $V_{w|V|\times|H|}$ — матрица весов слов, h'_t — модифицированный скрытый слой, где для всех $i \notin C$ $h_t(i) = 0$.

Таким образом, условные вероятности слов факторизуются в произведение вероятностей класса и слова в данном классе. Ускорение обучение достигается за счет того, что статистическая сумма софтмакс-функции и распределение, вычисленное на выходном слое задействуют существенно меньшее число элементов. Выигрыш по скорости оказывается пропорционален числу классов.

Стоит отметить, что сам Т.Миколов рекомендует использовать как можно меньшее количество классов (в идеале 1) [13]. Более того, выигрыш от использования классов оказывается не таким серьезным, если вычисления производятся на GPU.

Данная модель, тем не менее весьма популярна, в частности, она была использована в [72] для экспериментов с русским языком. Одной из причин популярности модели, помимо выигрыша в затратах на обучение при невозможности параллелизации, состоит в том, что модель реализована в рамках свободно распространяемой RNNLM Toolkit Т.Миколова [73].

2.6.2 Результаты рекуррентных нейронных сетей на корпусе Penn TreeBank

В [13] показано, что языковая модель, основанная на рекуррентной нейронной сети демонстрирует наилучшие среди всех моделей результаты на стандартном корпусе Penn TreeBank, являющемся подкорпусом корпуса Wall Street Journal с синтаксической разметкой. В своих экспериментах Т.Миколов использовал различные существующие языковые модели, начиная от n-граммных моделей с различным сглаживанием, заканчивая нейронными сетями с прямым распространением. Также проводились эксперименты с линейной интерполяцией моделей.

В ходе эксперимента словарь был ограничен до 10^5 слов. Объем обучающих данных составил $9.3 \cdot 10^5$ токенов.

Таблица 2.1 представляет собой фрагмент таблицы, приведенной в [13].

Из таблицы видно, что рекуррентная нейронная сеть (скрытый слой 300, глубина развертки 5) дает наилучший показатель перплексии на тестовом корпусе. Комбинация нескольких сетей (скрытый слой 100-500) позволяет уменьшить перплексию до уровня 102.1. Объединение всех моделей дает перплексию 83.5.

Распределение весов каждой из моделей при их интерполяции также свидетельствует о превосходстве рекуррентной модели: совокупный вес рекуррентных моделей составляет 0.63, наилучшая не нейросетевая модель — леса решающих деревьев — имеет вес лишь 0.1.

Таблица 2.1: Результаты экспериментов Т.Миколова на Penn TreeBank [13]

Модель	**Перплексия**
3-граммы, сглаживание Кнесера-Нея	148.3
5-граммы, сглаживание Кнесера-Нея	141.2
5-граммы, сглаживание Кнесера-Нея с кэшированием	125.7
Максимальная энтропия, 5-граммы	142.1
Леса решающих деревьев	131.9
Структурная модель	146.1
Нейронная сеть с прямым распространением	140.2
Синтаксическая модель на нейронной сети с прямым распространением	131.3
Рекуррентная нейронная сеть	124.7
Комбинация рекуррентных нейронных сетей	102.1

2.6.3 Проблема угасания градиента и моделирование языка

В [13] не предпринимается никаких попыток решения проблемы затухания градиента. Напротив, этот факт принимается как данность и из него делается оптимистический вывод о том, что оптимальной глубиной развертки является 6, поскольку далее на графике зависимости перплексии от глубины развертки наблюдается плато (рис. 2.3). Также отмечается, что даже без развертки сеть демонстрирует показатели перплексии сравнимые с 4-граммной моделью.

Тем не менее, достаточно легко найти пример фразы, успешное предсказание слов которой может быть осуществлено лишь с учетом более длинных зависимостей:

Студент московского ордена Ленина, ордена Октябрьской революции и ордена Трудового Красного знамени государственного университета им. М. В. Ломоносова получил красный диплом.

Для русского языка это представляется особенно актуальным ввиду необходимости учитывать согласование слов в предложении. В примере выше глаголы мужского рода прошедшего времени должны получать большую вероятность, чем, скажем, глаголы женского, однако рекуррентная нейронная сеть, скорее всего, будет не способна учесть этот факт.

С начала 2010-х попытки разработки моделей, устраняющих данный недостаток, предпринимались неоднократно.

Сам Т.Миколов в [28] пытается решить проблему учета дальних зависимостей добавлением информации, поступающей на входной слой: помимо стандартного вектора слова x_t на вход сети также подается вектор тем текущего документа, оцененный по доступному левому контексту алгоритмом LDA (Latent Dirichlet Allocation). LDA осуществляет вложение документа в векторное пространство: документ представляется как вектор в пространстве *тем*. Каждая координата полученного вектора представляет собой вероятность того, что скрытая тема присутствует в документе. Подробно об LDA в контексте тематического моделирования будет сказано в главе 3.

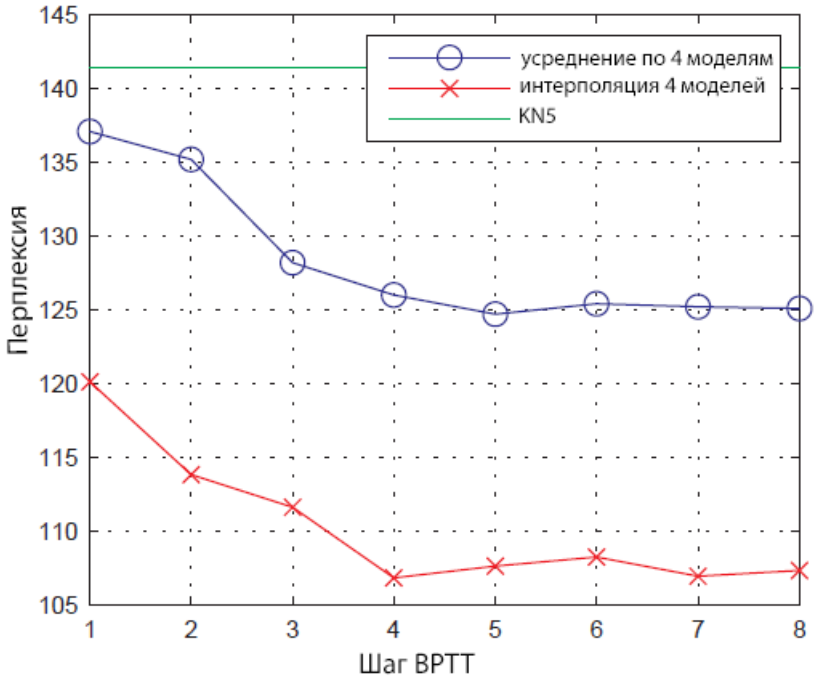

Рисунок 2.3: Зависимость качества модели на рекуррентной нейронной сети от глубины развертки для среднего результата по 4 моделям и для их интерполяции [13]

Таким образом, начиная с некоторой длины левого контекста, на вход сети помимо текущего слова подавалось также сжатое представление всего левого контекста, что могло компенсировать отсутствие явного моделирования дальних зависимостей.

Данный подход позволил получить выигрыш по перплексии 13.0 (с 124.7 до 113.7) и заметное снижение пословной ошибки (с 16.%6 до 14.6%) пунктов по сравнению с обычной рекуррентной архитектурой на Penn TreeBank и WSJ соответственно.

Прочие решения рассматривались прежде всего в контексте экспериментов с патологическими задачами моделирования сверхдлинных зависимостей. Здесь стоит отметить результаты, полученные с использованием LSTM [71].

В [62] тестируются различные эвристики и топологии, имеющие целью улучшить текущие наилучшие показатели в ряде тестов. Для моделирования языка на Penn TreeBank удается получить показатели перплексии 107.5 — наилучший опубликованный результат на Penn TreeBank.

2.7 Рекуррентные нейронные сети для моделирования флективных языков

При наличии словаря существенного объема статистическое моделирование флективных языков составляет дополнительную техническую проблему для нейросетевого подхода. Большое количество различных словоформ приводит к пропорционально большему размеру вы-

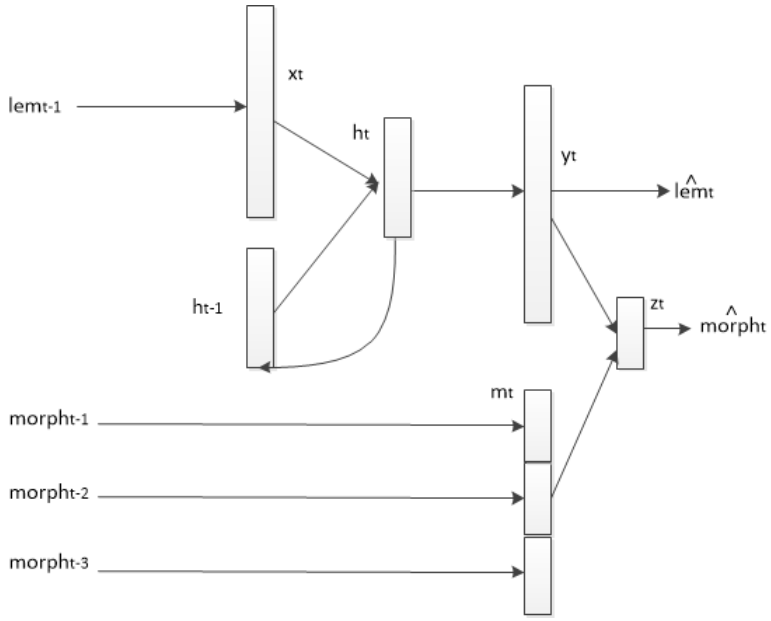

Рисунок 2.4: Рекуррентная нейронная сеть с внешним классификатором. m_t — конкатенация бинарных векторов с ненулевой координатой, соответствующей номеру вхождения грамматической пометки в списке допустимых пометок (словаре) GL; z_t — вектор длины $|GL|$ — оценка распределения грамматических форм для слова w_t

ходного слоя, а из (2.19) видно, что сложность алгоритма обучения линейна по объему выходного слоя.

Чтобы обойти эту проблему, можно было бы использовать схему на рис.2.4. Каждое входное слово предварительно лемматизуется внешним морфологическим анализатором. Леммы используются для предсказания последующих лемм. Далее для предсказанной леммы запускается линейный классификатор (например, логистическая регрессия), предсказывающий словоформу по лемме и морфологическим признакам контекста. Данный подход позволяет миновать проблему разрастания словаря. Другой подход мог бы состоять в том, чтобы разделить выходной слой на два вектора — словарный (леммы) и морфологический (морфологические признаки). Ошибка предсказания в данном случае получалась бы суммированием ошибок на двух векторах.

Подробно решение этой проблемы будет рассмотрено в главе 4. Ниже приводятся результаты предварительного эксперимента, целью которого являлась проверка гипотезы о том, что комбинация нейронных сетей, обученных на леммах, дает лучший результат, чем комбинация n-граммных моделей с дисконтированием Кнесера-Нея [3].

2.7.1 Описание экспериментов на лемматизованном корпусе

Целью эксперимента была проверка качества моделей на задаче ранжирования равновероятных гипотез распознавания речи на русском языке.

Модель была натренирована на новостном корпусе объемом приблизительно в $2 \cdot 10^6$ токенов. Примерно 10% данных было выделено для валидации. Каждый текст был обработан

морфологическим анализатором/лемматизатором для русского языка [74] со встроенным словарем примерно в $2 \cdot 10^6$ словоформ. Выходом анализатора являлся текст, в котором все известные словоформы были заменены соответствующими леммами, а неизвестные — специальным токеном "UNK". Второй сгенерированный текст был получен только заменами неизвестных токенов на "UNK". Таким образом, было получено 2 пары тренировочной и тестовой выборок для лемм и словоформ соответственно. На этих корпусах проводилось обучение и эксперименты по определению перплексии.

Во втором эксперименте оценивалось влияние моделей на ранжирование гипотез распознавания. Гипотезы генерировались внешней системой распознавания фирмы Nuance. К сожалению, декларируемый показатель WER данной системы, как и полный список гипотез, недоступен. Для генерации гипотез использовался подкорпус русскоязычного корпуса звучащей речи со студийным качеством записи. Подкорпус содержал высказывания на русском языке, записанные 20 дикторами: 10 мужских голосов и 10 женских. Для каждого диктора случайным образом выбиралось 80 аудиозаписей. Записи подавались на вход системе распознавания. На выходе получалось до 10 гипотез. В результате была получена коллекция неотсортированных списков гипотез. Как правило, список не содержал полностью правильной гипотезы, и она добавлялась вручную.

Далее каждая гипотеза обрабатывалась теми же инструментами, которые использовались при подготовке корпусов: т.е. были проведены лемматизация и замены неизвестных слов. Полученные корпуса были обработаны обученными на предыдущем этапе моделями. В результате для каждой из гипотез были получены списки откликов от каждой модели — n-граммной со сглаживанием Кнесера-Нея и рекуррентных нейронных сетей с различными размерами скрытого слоя. Всего в обучающем корпусе для ранжирования было 1300 фраз со средним значением 5 гипотез на фразу. В тестовом корпусе было 300 фраз.

В тестах были использованы n-граммные модели со сглаживанием Кнесера-Нея, порядков 3,4,5, натренированные на леммах и на словоформах. Модели на основе рекуррентных сетей различались размером скрытого слоя. Были протестированы модели с объемами слоя 100,200,300, 400 и 500. Все рекуррентные сети обучались на лемматизованном корпусе. Кроме того, использовалась оценка, возвращаемая морфологическим анализатором. В результате было получено 12 оценок. Для ранжирования использовалась модель ranking SVM [75], где в качестве признаков использовались оценки моделей. Результирующая модель обучалась ранжированию гипотез в списке на 2 категории — верная и неверная гипотеза. Фактически, данный подход дает интерполяцию моделей. В качестве метрик для оценки в этом случае выбраны уровень пословной ошибки (word error rate, WER%) и процент случаев выбора правильной гипотезы (sentence error rate, SER%).

2.7.2 Результаты экспериментов на лемматизованном корпусе

Результаты экспериментов приводятся в таблицах 2.2 и 2.3. В таблице 1 приведены перплексии всех используемых моделей. В таблице 2 приведены результаты эксперимента по ранжированию — уровень пословной ошибки (WER%) и точность выбора правильной гипотезы (SER%).

Таблица 2.2: Перплексии моделей на тестовой выборке

Модель	**Перплексия**	**Модель**	**Перплексия**
KN3$_{lem}$	272.8	**RNN100**	240.13
KN4$_{lem}$	272.2	**RNN200**	230.45
KN5$_{lem}$	273	**RNN300**	231
KN3$_{tok}$	128.72	**RNN400**	231.87
KN4$_{tok}$	130.76	**RNN500**	231.21
KN5$_{tok}$	132		

Стоит отметить, что перплексии моделей, натренированных на лемматизованном и нелемматизованном корпусе, строго говоря, не сравнимы по перплексии, поскольку количество неизвестных токенов, а значит и словарный состав корпусов, различны. Таким образом, важным обнадеживающим выводом, который можно сделать из приведенной таблицы, является то, что модели на рекуррентных нейронных сетях демонстрируют существенно лучшие показатели в эксперименте, чем 5-граммная модель со сглаживанием Кнесера-Нея.

Таблица 2.3: Результаты моделей в эксперименте по ранжированию

Модель	WER%	SER%	**Модель**	WER%	SER%
KN5$_{lem}$	16.62	40.8	**RNN100**	17.55	43.67
KN5$_{tok}$	18.09	42.72	**RNN200**	15.35	40.5
KN5$_{lem}$ **+ morph**	15.58	43.98	**RNN300**	17.09	43.98
KN$_{lem}$ **all**	17.05	40.82	**RNN400**	16.58	41.77
KN$_{lem}$ **all + morph**	15.74	43.67	**RNN500**	17.43	43.67
KN$_{lem+tok}$ **all**	15.74	39.24	**RNN all**	**15.35**	**38.29**
KN$_{lem+tok}$ **all + morph**	15.89	43.35	**RNN all + morph**	14.58	41.45
все модели	14.78	40.5			

Рассмотрим теперь результаты эксперимента по ранжированию. Стоит сделать следующие замечания. Первое из них состоит в заметном превосходстве рекуррентных нейронных сетей над сглаженными n-граммами. Второй заметный факт — это противоречивое влияние морфологической модели на конечный результат: улучшение пословной ошибки при явной тенденции к голосованию за неверную гипотезу предложения. Это можно объяснить тем фактом, что оценка, возвращаемая морфологическим анализатором, пропорциональна вероятности лучшего разбора $P(tag_1^T|word_1^T)$. По этой причине данная оценка имеет тенденцию к выбору гипотез с наименьшей энтропией разбора. Стоит признать, что данная оценка не вполне подходит к решаемой нами задаче. Третий заметный факт состоит в несколько хаотичном характере результатов рекуррентных моделей: некоторые из них демонстрируют

достаточно скромные результаты, однако их интерполяции обеспечивают наилучшие результаты.

Эксперименты по ранжированию в целом демонстрируют превосходство рекуррентных моделей. Наилучшая комбинация задействует оценку, возвращаемую морфологическим анализатором и оценки, полученные от рекуррентных моделей. Таким образом, обеспечивается комбинирование морфологической и словарной информации. Данный результат свидетельствует о том, что исследования в данном направлении могут быть продолжены.

2.7.3 Эксперименты на корпусе без лемматизации

Помимо уже упомянутых экспериментов с чешским языком, результаты которых стоит признать не вполне надежными, стоит отметить работу [72], в которой описан эксперимент с применением рекуррентной нейронной сети на русском корпусе. В статье проводится сопоставление двух *продвинутых* (advanced) языковых моделей — собственно рекуррентной нейронной сети и факторной языковой модели, разработанной авторами.

Набор факторов в модели, используемой авторами был получен при помощи процедуры отбора признаков, основанной на генетическом алгоритме ([76]).

Эксперименты проводились на достаточно большом корпусе, объемом $41 \cdot 10^6$ токенов. Используемый словарь составил $100 \cdot 10^4$ слов. Корпус был составлен на основе материалов русскоязычных новостных интернет-ресурсов, опубликованных в период 2006-2011 гг. В качестве тестового множества использовалась часть объемом 10^6 токенов. С другой стороны, в ряде экспериментов, обучение проводилось на подкорпусе объемом в $10 \cdot 10^6$ токенов.

Используемые факторной моделью признаки (род, число, падеж и т.д.) были получены при помощи внешних свободных утилит для обработки текста [77].

Эксперименты по распознаванию речи проводились на корпусе СПИИРАН [78] и корпусе GlobalPhone [72]. В общей сложности корпус содержал 28671 высказывание при количестве дикторов равном 65. Общая длительность корпуса составила 38 часов. В тестовую выборку было выделено 10% корпуса GlobalPhone.

Факторная модель была создана с использованием свободно распространяемого пакета для языкового моделирования SRILM. Модель, основанная на рекуррентной нейронной сети, была обучена и протестирована при помощи RNNLM Toolkit — утилиты Т.Миколова [73].

Из доступных вариантов в RNNLM Toolkit авторы выбрали архитектуру с большим количеством классов, что объясняется размерами обучающего корпуса. Количество классов, выбранное авторами (300) весьма велико, и если принять во внимание рекомендации Т.Миколова, — не оптимально. Авторы используют готовую утилиту и не задействуют возможность параллелизации вычислений, что существенно затрудняет возможность оптимизации параметров — в данном случае числа классов и размеров скрытого слоя.

Результаты, показанные рекуррентной архитектурой в экспериментах [72], оказываются существенно хуже 3-граммной модели с сглаживанием Кнесера-Нея, взятой в качестве базо-

Таблица 2.4: Результаты интерполяции RNNLM и KN3. λ — коэффициент интерполяции рекуррентной модели [72]

Скрытый слой	**Количество классов**								
	150			**500**			**100**		
	λ	**PPL**	**WER**	λ	**PPL**	**WER**	λ	**PPL**	**WER**
1000	0.68	466	33.5	0.64	470	33.3	0.68	503	33.3
150	0.67	474	33.6	0.67	457	33.3	0.66	462	33.1
200	0.69	454	33.3	0.77	458	33.5	0.64	471	34.1
250	0.82	459	33.9	0.63	469	33.5	0.67	459	33.5

вой. Модель, натренированная на корпусе объемом 10^6 токенов, демонстрирует перплексию $PPL = 1100$ и уровень пословной ошибки $WER = 38\%$ против 537 и 35.4% соответственно у KN3.

Ненамного лучше оказываются и результаты интерполяции RNNLM и KN3 (см. таблицу 2.4):

Как видно из таблицы, на этот раз показатели превзошли результаты KN3 и существенно превзошли RNNLM, запущенную в одиночку, однако результаты нельзя назвать впечатляющими. Более того, они не идут ни в какое сравнение с результатами для английского языка на Penn TreeBank. Из таблицы можно было бы сделать вывод, что результат не зависит от количества классов, однако логичнее предположить, что это зависимость просто перестает быть заметной при существенно меньшем их количестве (меньше 100).

Тем не менее, даже притом, что в данном случае вполне можно предположить, что гиперпараметры модели были не оптимальны, основной причиной столь скромного результата, по-видимому, стоит считать тот факт, что стандартная архитектура выходного слоя не приспособлена для флективных языков, что доказывает уже высказанный ранее тезис.

Тем не менее, результаты, полученные путем интерполяции KN3 с факторной моделью, оказываются ненамного лучше: так, наилучшая конфигурация факторной модели дает $WER = 33\%$, наилучшую перплексию — 437. Разница перплексий заметна, однако, как отмечалось ранее, это не сказывается серьезно на качестве распознавания, что, собственно и видно при сопоставлении уровней пословной ошибки.

Таким образом, даже в условиях, когда рекуррентная модель не должна давать серьезных улучшений — обучение на словоформах и неоптимальные гиперпараметры, — она практически не уступает модели, специализированной для флективного языка.

2.8 Выводы

В данной главе были рассмотрены следующие вопросы:

– описание метода векторного представления слов на при помощи нейронных сетей.

– архитектура рекуррентных нейронных сетей и алгоритм распространения ошибки обратно по времени.

– принципиальные ограничения алгоритма распространения ошибки обратно по времени и методы их обхода.

– ограничения алгоритма распространения ошибки обратно по времени в контексте статистического моделирования языка.

Основным результатом данной главы является экспериментальное подтверждение эффективности языковой модели на рекуррентной нейронной сети для моделирования русского языка без учета морфологии. Эксперимент демонстрирует, что проблема свободного порядка слов не является существенной для данной модели. Рекуррентная модель оказывается более эффективной, чем 5-граммная модель со сглаживанием Кнесера-Нея. Преимущество наблюдается как в эксперименте по измерению перплексии, так и в эксперименте по ранжированию гипотез распознавания.

Кроме того, была предложена модель с внешним классификатором морфологических форм.

Таким образом, принимая во внимание также факты, приведенные в настоящей главе, можно предположить, что улучшение качества предсказания может идти следующими путями:

а) совершенствование процедуры обучения (использование аддитивной регуляризации, методов второго порядка, ограничений амплитуды градиента);

б) использование специализированной математической модели памяти, не подверженной проблеме угасания градиента (LSTM);

в) комбинирование рекуррентной сети с другими моделями, учитывающими контекст (PLSA, LDA).

На основании приведенных выше утверждений можно сделать следующий вывод: рекуррентные нейронные сети являются перспективным инструментом для моделирования флективных языков. Тем не менее, исходная архитектура должна быть модифицирована для обеспечения поддержки большого словаря. Другая обозначенная проблема, а именно учет длинного контекста, — может быть решена с использованием тематического моделирования.

В следующих главах будут рассмотрены модификации рекуррентной нейронной сети, отвечающие поставленным требованиям.

Глава 3

Расширение рекуррентной архитектуры с помощью тематического моделирования

3.1 Введение

В предыдущей главе было показано, что стандартная архитектура рекуррентной нейронной сети не способна учитывать дальние зависимости между элементами входной последовательности. В данной главе будет предложен способ уменьшения эффекта угасания градиента за счет использования векторного представления всего доступного левого контекста, основанного на вероятностном тематическом моделировании.

Глава структурирована следующим образом. В разделе 2 будет кратко описана и проанализирована предлагаемая модель. Раздел 3 посвящен задаче тематического моделирования, рассмотрены методы ее решения и основные алгоритмы. В разделы 4–5 подробно описаны методы тематического моделирования, используемые далее. В разделе 6 приводятся результаты экспериментов. Наконец, в разделе 7 делаются основные выводы.

3.2 Предлагаемая модель

3.2.1 Векторные представления слов

Идея о векторном представлении слов, т.е. сопоставлении словам из словаря $\mathbb{V}$ некоторого вектора относительно небольшой размерности $d \ll |\mathbb{V}|$ насчитывает несколько десятилетий, начиная с известной работы [79], в которой был описан латентно-семантический анализ на основе сингулярного разложения матрицы термин–документ. Основанием практически всех используемых векторных вложений слов является интуитивное представление о том, что смысл слов определяется контекстами, в которых данное слово может употребляться. Помимо уже упомянутых моделей латентно-семантического анализа — основанных на сингулярном или неотрицательном ([26]) матричном разложении — существуют также модели, основанные на нейронных сетях ([58]), уже упоминавшиеся в главе 1.

В 2013 году Т.Миколов представил два новых способа получения векторных вложений ([80] [81]). Первый из них базировался на языковой модели на рекуррентной нейронной сети, описанной в предыдущей главе. В качестве векторных представлений закономерно были использованы столбцы матрицы U, первого слоя. Второй подход был предложен в [82], где было экспериментально показано, что векторные представления, основанные на нейронных сетях — как рекурсивных, так и с прямым распространением — уступают в задаче определения семантического сходства довольно простой лог-линейной модели (т.н. n-грамм с пропуском, *skip-gram model*).

В skip-gram моделируется распределение пар слов в контексте длины l: $\mathsf{P}(w_k|w_t)$, $0 < |t-k| \leq l$.

$$P(w_k|w_t) = softmax(v_w(w_t) \cdot v_c(w_k)),$$

где $v_w(w_t), v_c(w_k) \in \mathbb{R}^d$ — векторные представления предикторного w_t и целевого w_k слов. Таким образом, каждое слово w обучающего корпуса порождает $2l$ пар в обучающей выборке D. Важным предположением данной модели является то, что для одного и того же слова словаря $w \in \mathbb{V}$ векторные представления $v_w(w)$ и $v_c(w)$ будут различны. Это отвечает представлению о том, что одно и то же полнозначное слово редко будет встречаться в достаточно коротком контекстном окне длины l. В противном случае поскольку $v_w^T v_w \geq v_w^T v_c$ редко наблюдаемые последовательности типа *собака собака собака* всегда получали бы большую вероятность в модели. В оригинальной статье было продемонстрировано, что геометрическая близость векторных представлений v_w хорошо согласуется с интуитивным представлением о семантической близости слов.

Ввиду вычислительной сложности (необходимость считать статистическую сумму в знаменателе софтмакс-функции по всем словам) в описанном выше виде данная модель не используется. Вместо нее была предложена модель с так называемой *иерархической софтмакс-функцией* [81].

Удивительным свойством описанной модели являлось то, что для слов ряда категорий (особенно в этой связи известны примеры со столицами стран) выполнялись интуитивно понятные соотношения:

$$v_{Париж} - v_{Франция} \approx v_{Рим} - v_{Италия}$$

Данную модель в сообществе исследователей векторных моделей по непонятной причине принято называть нейросетевой [83] и даже более того, основанной на глубоком обучении, хотя стоит отметить, что ни в одной из статей [80], [81], [82] данная модель не называется нейросетевой. Причиной как данного весьма распространенного заблуждения, так и феноменальной популярности самой модели, по всей вероятности, является ее анонсирование под брендом компании Google в виде пакета *word2vec* в 2013 году. Связь с компанией, известной своими экспериментами с глубоким обучением, стала причиной взрывной популярности идеи

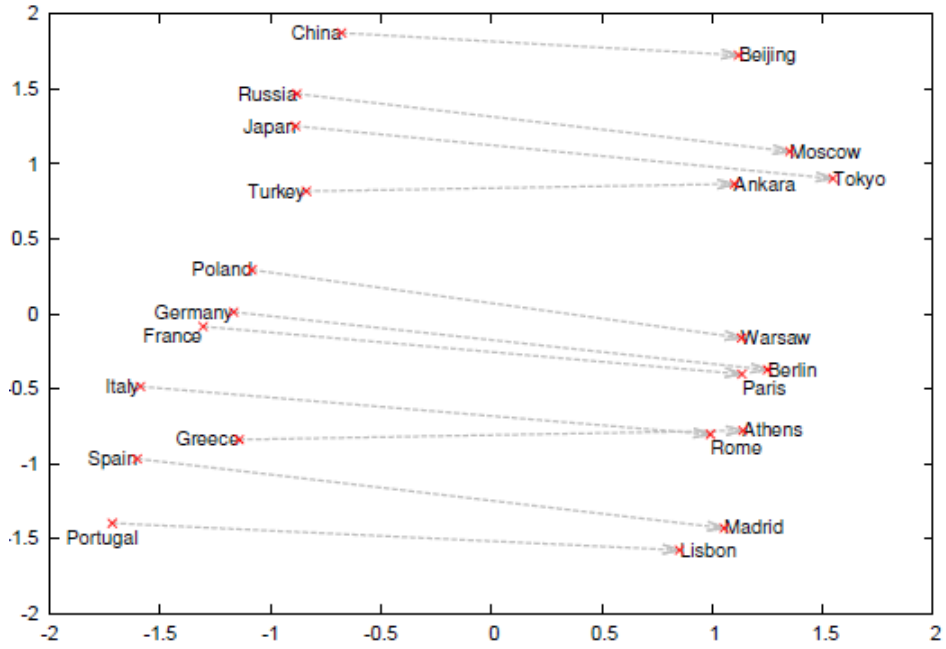

Рисунок 3.1: PCA-проекция векторов, соответствующих именам столиц, демонстрирует линейные соотношения между векторными вложениями слов [80]

векторных представлений слов, причем сравнительно небольшое число работ было посвящено собственно анализу моделей, лежащих в основе word2vec [84], [85].

Анализ модели skip-gram, проведенный в [85], выявил ее сходство с уже известными моделями латентно-семантического анализа.

Целевая функция модели выглядит следующим образом ([84]):

$$\mathcal{L}(\theta) = \sum_{(w,c)\in D} \log \sigma(v_c^{(\theta)} \cdot v_w^{(\theta)}) + \sum_{(w,c)\in D'} \log \sigma(-v_c^{(\theta)} \cdot v_w^{(\theta)}) \tag{3.1}$$

В уравнении выше D — множество пар слово-контекст в обучающем корпусе, D' — множество полученное путем семплирования k контекстов для каждого из слов w обучающего корпуса. k является гиперпараметром для процедуры обучения. Семплирование контекстов длины L производится на основе униграммной модели. Таким образом, математическое ожидание функции правдоподобия в уравнении 3.1 можно переписать в виде:

$$\mathbb{E}_D\mathcal{L}(\theta) = \sum_{w\in V}\sum_{c\in V^L} N(w,c)\log\sigma(v_c^{(\theta)}\cdot v_w^{(\theta)}) + \sum_{w\in V}\sum_{c\in V^L} N(w)\cdot k\cdot \mathbb{E}_C \log\sigma(-v_c^{(\theta)}\cdot v_w^{(\theta)}) =$$
$$\sum_{w\in V}\sum_{c\in V^L} N(w,c)\log\sigma(v_c^{(\theta)}\cdot v_w^{(\theta)}) + \sum_{w\in V} N(w)\cdot k \sum_{c\in V^L}\frac{N(c)}{|D|}\log\sigma(-v_c^{(\theta)}\cdot v_w^{(\theta)}) \tag{3.2}$$

Тогда каждая наблюдаемая пара «слово-контекст» вносит вклад равный

$$l(w,c;\theta) = N(w,c)\log\sigma(v_c^{(\theta)}\cdot v_w^{(\theta)}) + N(w)\cdot k\frac{N(c)}{|D|}\log\sigma(-v_c^{(\theta)}\cdot v_w^{(\theta)})$$

Взяв производную $\frac{\partial l}{\partial(w \cdot c)}$ и приравнивая нулю, получаем решение относительно $w \cdot c$:

$$w \cdot c = \log(\frac{N(w,c) \cdot |D|}{N(w)N(c)}) - \log k = PMI(w,c) - \log k$$

Таким образом, максимизация функционала 3.1 равносильна поиску разложения матрицы поточечных взаимных информаций между словами w_i и контекстами c_j.

Сведение векторных вложений к матричным разложениям возможны и для других моделей [85].

Несмотря на то, что сделанный выше вывод не объясняет всех наблюдаемых свойств векторных вложений слов, он устанавливает связь моделей word2vec с одним из самых успешных и теоретически обоснованных методов вложения — латентным размещением Дирихле (LDA) и его обобщениями. На сегодняшний день LDA и другие модели вероятностного тематического моделирования лежат в основе многих промышленных решений в области анализа данных и извлечения информации.

Более того, в главах 1 и 2 уже упоминалось, что векторные представления слов и документов используются в статистическом моделировании языка. Поскольку тематическая модель позволяет получить «усредненное» представление левого контекста, если рассмотреть его как документ, потенциально это может компенсировать проблему угасания градиента в рекуррентной сети, речь о которой шла в предыдущей главе.

Ниже будет описана языковая модель, задействующая векторное представление левого контекста, полученное с помощью латентного размещения Дирихле.

3.2.2 Компенсация эффектов угасания градиента при помощи тематического моделирования

В главе 1 уже шла речь о некоторых расширениях n-граммной модели, направленных на преодоление присущих ей недостатков, связанных с ограниченностью контекста. Так, речь шла о моделях с кэшированием [30], а также моделях, использующих латентно-семантический анализ (LSA) [24], [38]. Левый контекст в моделях с LSA представляется в виде вектора, а предсказание основывалось на косинусном расстоянии между векторным представлением контекста и векторным преставлением слова-кандидата. Другим способом расширения контекста являются модели, основанные на классификации левых контекстов по темам. При дальнейшем предсказании предсказанная тема становится дополнительным предиктором для слова [86], [32]. Как правило, признаки, основанные на длинных контекстах, используются в моделях максимальной энтропии [87], [88].

В 2012 году Т.Миколов опубликовал работу [28], в которой расширил предложенную им ранее рекуррентную нейронную сеть для моделирования языка, добавив туда дополнительный вектор признаков, получаемый путем тематического разложения левого контекста методом LDA. При этом вектор LDA поступал как на вход скрытого слоя, так и на вход выходного

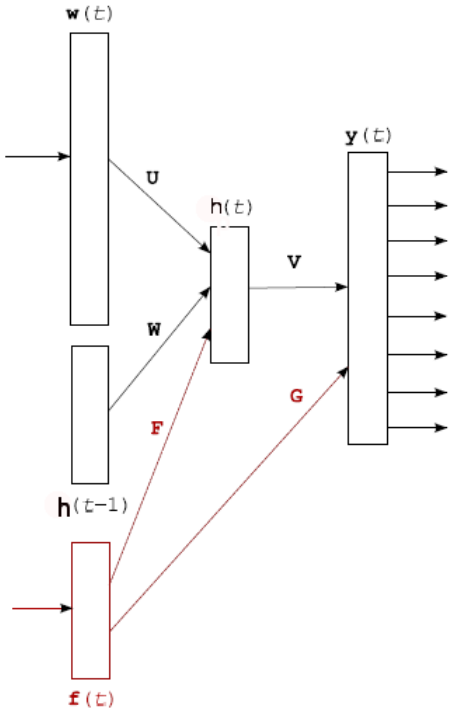

Рисунок 3.2: Языковая модель на рекуррентной нейронной сети с использованием латентного размещения Дирихле. f_t — тематическое представление левого контекста на основе LDA на шаге t; F — матрица весов тематических признаков; G — матрица «прямых связей» тематического разложения и выходов сети [28]

слоя (см. рис. 3.2). Никаких теоретических обоснований именно такой архитектуры авторы не приводят.

Таким образом, стандартная модель RNNLM принимает следующий вид:

$$h_t = \sigma(W \cdot h_{t-1} + U \cdot x_t + F \cdot f_t)$$

$$y_t = softmax(V \cdot h_t + G \cdot f_t),$$

где f_t — LDA-разложение левого контекста, G и F — матрицы весов.

В качестве векторов LDA, впрочем, в [28] авторы используют не вектор распределения тем $\theta = p(\mathbf{t}|d)$, с левым контекстом в качестве документа d, а вектор $\theta' = p(\mathbf{t}|w)$, полученный из LDA разложения по теореме Байеса. Приближение вектора θ получается путем перемножения векторов $\theta'_{w(t)}$ с последующей перенормировкой:

$$f_t = \frac{1}{Z} f_{t-1}^{\gamma} \theta'^{1-\gamma}_{w(t)},$$

где γ -весовой коэффициент, $\theta'_{w(t)} = p(\mathbf{t}|w)$, Z - нормировочный коэффициент.

Используя данную модель, в тестах на корпусе Penn TreeBank авторы получают результат 113.7 по перплексии, что на 8.8% лучше результата изолированной рекуррентной сети. В целом данный результат близок к рекордным (см. главу 2). Примечательно, что замена LDA

на классический LSA, основанный на сингулярном разложении, дает даже немного лучший результат: 110.3. Данное различие, впрочем, нельзя назвать существенным.

В данном контексте довольно интересным кажется проанализировать влияние различных тематических моделей на конечный результат, не ограничиваясь только LDA. В частности, выявить эффекты от разреживания и выделения стоп-слов [39].

Экспериментирование с различными аддитивными регуляризаторами может привести к улучшениям относительно модели, основанной на латентном размещении Дирихле. В следующем разделе дается подробное описание аппарата тематического моделирования, используемого в дальнейших экспериментах.

3.3 Вероятностное тематическое моделирование

Тематическое моделирование является одним из современных направлений анализа текста и документов. Базовой идеей тематического моделирования является предположение о наличии в документах ограниченного набора скрытых порождающих «тем». Каждая из тем «ответственна» за появление в тексте определенной, присущей ей лексики. Собственно *тематическая модель* (topic model) коллекции текстовых документов определяет, к каким темам относится каждый документ и какие слова (термины) образуют каждую тему.

Как уже говорилось ранее, тематическая модель может быть основана на сингулярном разложении матрицы частот слов в документах (матрица «термин-документ») $W = \Phi\Theta$. В этом случае полученные матрицы Φ и Θ, вообще говоря, могут иметь отрицательные элементы, что делает такие модели зачастую неинтерпретируемыми.

С другой стороны, разложение матрицы «термин-документ» может быть произведено так, что полученные матрицы Φ и Θ будут стохастическими, а их столбцы будут представлять собой дискретные вероятностные распределения. Таким образом, вероятностная тематическая модель описывает каждую тему дискретным распределением на множестве терминов, а каждый документ — дискретным распределением на множестве тем [39]. Данный подход к латентно-семантическому анализу называется вероятностным латентно-семантическим анализом (PLSA).

Идея PLSA была предложена Т.Хоффманом в [89]. Вероятностное тематическое моделирование также можно рассматривать как «мягкую» кластеризацию документов и слов по кластерам, роль которых выполняют темы. Мягкая кластеризация лексики представляет отдельный интерес в различных областях обработки текста. В то же время тематические модели находят применение в классификации документов [90], изображений и видео, в информационном поиске [91] и рекомендательных системах [92].

Огромным преимуществом вероятностного тематического моделирования является возможность учета различных факторов и наложения на модель специфических ограничений в зависимости от решаемой задачи, что делает аппарат вероятностного тематического моделирования удобным в том числе и для задачи предсказания слов по контексту.

3.3.1 Формальная постановка задачи тематического моделирования

Будем рассматривать множество документов D, на языке со словарем $\mathbb{V}$. Тогда каждый документ $d \in D$ можно представить в виде $|\mathbb{V}|$-мерного вектора $(f(w_1), f(w_2), \ldots, f(w_{|\mathbb{V}|}))$, где $f(w)$ — частота вхождений слова $w \in \mathbb{V}$ в документ d.

Будем интерпретировать частоты $f(w,d)$ как оценки совместной вероятности $p(w,d)$ пары термин-документ в коллекции.

Далее примем гипотезу условной независимости слова от документа при известной теме:

$$p(w|d,t) = p(w|t)$$

Тогда вероятность $p(w,d)$ можно представить как:

$$p(w,d) = \sum_{t \in T} p(w|t)p(t|d), \tag{3.3}$$

где T — множество тем в модели. Вероятности $p(w|t)$ и $p(t|d)$ являются параметрами тематической модели и определяются на стадии обучения [93].

Если количество тем $|T|$ не превышает объема словаря и количества документов в коллекции, то используя 3.3 можно получить разложение матрицы «термин-документ»:

$$F \approx \Phi\Theta, \tag{3.4}$$

где $F = (p(w,d))_{|V| \times |D|}$, $\Phi = (p(w|t))_{|V| \times |T|}$, $\Theta = p(t|d)_{|T| \times |D|}$.

В отличие от классического LSA, для получения которого решается задача минимизации нормы Фробениуса для разности матриц $||\Phi\Theta - F||^2$, для вероятностного латентно-семантического разложения используются оценки максимального правдоподобия элементов матриц Φ и Θ, что эквивалентно минимизации дивергенции Кульбака-Лейблера между эмпирическими оценками вероятностей $\hat{p}(w|d) = \frac{C(w,d)}{\sum_w C(w,d)}$ и вероятностью в текущей тематической модели $p(w,d) = \sum_{t \in T} p(w|t)p(t|d)$:

$$\sum_{d \in D} C(d) KL(\frac{C(w,d)}{C(d)} || \phi_{w,t}\theta_{t,d}) \to min,$$

где $C(d) = \sum_w C(w,d)$, $\phi_{w,t} = p(w|t)$, $\theta_{t,d} = p(t|d)$.

Дивергенция Кульбака-Лейблера рассчитывается по следующей формуле:

$$KL(P||Q) = \sum_{i=1}^{n} p_i \ln \frac{p_i}{q_i}$$

Известно, что минимизация дивергенции Кульбака-Лейблера между эмпирическим распределением P и семейством теоретических распределений Q по параметрам Q эквивалентна максимизации правдоподобия для Q [93].

Таким образом задача подбора параметров тематической модели сводится к задаче матричного разложения, минимизирующего взвешенную сумму дивергенций Кульбака-Лейблера, где роль веса выполняет длина документа.

Рассмотрим теперь, каким образом осуществляется данное разложение.

3.3.2 Обучение тематической модели с помощью EM-алгоритма

Хоффман в [89] предложил использовать EM-алгоритм [94] для решения задачи матричного разложения 3.4. E-шаг состоит в в вычислении матожидания тем t в выборке n_{dwt} по текущим значениям параметров Φ и Θ:

$$\hat{n}_{dwt} = C(w,d)\frac{\phi_{wt}\theta_{td}}{\sum_{s\in T}\phi_{ws}\theta_{sd}} \tag{3.5}$$

На M-шаге полученные матожидания используются для обновления матриц Φ и Θ:

$$\phi_{wt} = \frac{\sum_{d\in D}\hat{n}_{dwt}}{\sum_{w\in V}\sum_{d\in D}\hat{n}_{dwt}} \tag{3.6}$$

$$\theta_{td} = \frac{\sum_{w\in V}\hat{n}_{dwt}}{N_d}, \tag{3.7}$$

где N_d — число документов в коллекции.

EM-алгоритм состоит в итеративном обновлении псевдо-наблюдений и параметров модели по формулам 3.5, 3.6 и 3.7. Алгоритм линеен по длине коллекции N_d, числу тем $|T|$ и числу итераций.

Обновление параметров ϕ_{wt} и θ_{td} на M-шаге требует прохода по всем документам $d \in D$ и всем словам $w \in d$. Переменные $\hat{n}_{dwt}$ фактически могут быть вычислены на M-шаге, когда они понадобятся для обновления параметров. Плюсом такого подхода является отсутствие необходимости хранения трёхмерной матрицы $\hat{n}_{dwt}$.

Есть, однако, ряд соображений, приводящих к еще более эффективным модификациям EM-алгоритма.

Модифицированный алгоритм, называемый *пакетным EM-алгоритмом для вероятностной тематической модели* приведен в листинге ниже.

В зависимости от выбранной стратегии инициализации θ_{td} на шаге 5 алгоритма 3.3.1 можно либо отказаться от хранения матрицы θ_{td} или использовать ее для инициализации распределения на последующих итерациях.

Скорость сходимости зависит от выбора числа итераций на внутреннем цикле по документу и внешнем цикле по коллекции.

Дополнительное ускорение сходимости можно получить, если начальные итерации провести не по всей коллекции, а по случайному подмножеству документов $D' \subseteq D$. В [93] утверждается, что данный прием работает хорошо на достаточно больших коллекциях.

Algorithm 3.3.1 Пакетный EM-алгоритм для PLSA

InitPhi() — функция начальных приближений матрицы «термин-тема» **InitTheta()** — функция начальных приближений матрицы «тема-документ» **InitZero()** — функция инициализации массива нулевыми элементами

Вход: коллекция документов D, количество тем $|T|$

Ф $\leftarrow InitPhi()$
повторять
  $\forall w \in V, t \in T : \hat{\mathbf{n}}_{wt}, \hat{\mathbf{n}}_t \leftarrow InitZero()$
  для $d \in D$
    $\theta_{td} \leftarrow InitTheta()$
    повторять
      $\forall w \in d : Z_w \leftarrow \sum_{t \in T} n_{dwt} \phi_{wt} \theta_{td}$
      $\forall t \in T : \theta_{dt} \leftarrow \frac{1}{n_d \sum_{w \in d}} n_{dw} \phi_{wt} \theta_{td} / Z_w$
    пока θ_{td} не сойдется
    $\hat{n}_{wt} \leftarrow \hat{n}_{wt} + n_{dw} \phi_{wt} \theta_{td} / Z_w$
    $\hat{n}_t \leftarrow \hat{n}_t + n_{dw} \phi_{wt} \theta_{td} / Z_w$
  $\forall t \in T, w \in V : \phi_{wt} \leftarrow \frac{\hat{n}_{wt}}{\hat{n}_t}$
пока Ф не сойдутся

Алгоритм 3.3.1 будет использован ниже для получения тематической модели, генерирующей векторные вложения левого контекста.

3.3.3 Разложение документа на основе существующей тематической модели

По мере обработки документа статистическая языковая модель должна вычислять новые вложения левого контекста в пространство тем, т.е. осуществлять тематическое разложение левого контекста.

Данное разложение может быть также получено путем запуска пакетного EM-алгоритма на коллекции из одного документа (весь доступный левый контекст) без обновления матрицы Ф. Фактически данная методика была предложена еще Т.Хоффманом [89]: фиксировать матрицу Ф, найденную по всем предыдущим документам, и определять только вектор θ_d для нового документа.

Эта эвристика основана на предположении, что коллекция достаточно велика, и один документ d не может существенно повлиять на оценки распределений ϕ_t. Вообще говоря, данное предположение может не выполняться, если документ d содержит значительное число новых терминов или относится к темам, слабо представленным в коллекции. Однако, в целом для обработки корпуса скользящим окном заданного объема, предположение Хоффмана можно принять.

3.4 Расширения вероятностного латентно-семантического анализа

3.4.1 Регуляризация тематических моделей

Как уже говорилось выше, неотрицательное (стохастическое) матричное разложение $\Phi\Theta$ не является единственным и определено с точностью до невырожденного преобразования:

$$\Phi\Theta = (\Phi S)(S^{-1}\Theta)$$

при условии, что матрицы $\Phi' = \Phi S$ и $\Theta' = S^{-1}\Theta$ также стохастические. Решаемая задача стохастического матричного разложения является некорректно поставленной. Неединственность решения влечёт неустойчивость EM-алгоритма: использование случайной инициализации приводит к тому, что для двух различных запусков алгоритма на одних и тех же данных полученные разложения будут отличаться достаточно сильно [93].

Общий подход к решению некорректно поставленных задач заключается во введении некоторых дополнительных ограничений —регуляризаторов — на параметры Φ, Θ, сужая тем самым множество решений. Ниже будет рассмотрен метод аддитивной регуляризации, когда оптимизируемый целевой функционал представляет собой линейную комбинацию исходного функционала качества L и регуляризаторов R_i, $i = 1 \dots n$ с неотрицательными коэффициентами регуляризации τ_i. Как и в исходной модели, многокритериальная оптимизация проводится при условии неотрицательности и нормировки столбцов матриц Φ и Θ:

$$L(\Phi, \Theta) + \sum_{i=1}^{n} \tau_i R_i(\Phi, \Theta) \to \max_{\Phi,\Theta}$$

В результате получим модифицированные формулы для M-шага в EM-алгоритме:

$$\phi_{wt} = \frac{\left(\sum_{d \in D} \hat{n}_{dwt} + \phi_{wt} \frac{\partial R}{\partial \phi_{wt}}(\Phi,\Theta)\right)_+}{\left(\sum_{w' \in V}\left(\sum_{d \in D} \hat{n}_{dw't} + \phi_{w't} \frac{\partial R}{\partial \phi_{w't}}(\Phi,\Theta)\right)\right)_+} \tag{3.8}$$

$$\theta_{dt} = \frac{\left(\sum_{d \in D} \hat{n}_{dwt} + \theta_{dt} \frac{\partial R}{\partial \theta_{dt}}(\Phi,\Theta)\right)_+}{\sum_{t' \in T}\left(\left(\sum_{w \in V} \hat{n}_{dwt'} + \theta_{dt'} \frac{\partial R}{\partial \theta_{dt'}}(\Phi,\Theta)\right)\right)_+} \tag{3.9}$$

Поскольку используется аддитивная регуляризация, добавление нового регуляризатора приводит к появлению нового слагаемого в вычислении $\frac{\partial R}{\partial \phi_{wt}}$ и $\frac{\partial R}{\partial \theta_{td}}$ в формулах M-шага 3.8 и 3.9. Благодаря этому свойству, различные тематические модели, сочетающие в себе большое число дополнительных требований, можно строить достаточно просто: необходимо лишь явно выписать формулу для нужного регуляризатора и его производную [95].

Ниже будут рассмотрены регуляризаторы, используемые в дальнейших экспериментах.

3.4.2 Сглаживающий регуляризатор

Сглаживающий регуляризатор минимизирует дивергенцию Кульбака-Лейблера между неизвестным распределением ϕ_{wt} или θ_{td} и некоторым априорным распределением β_w или α_t:

$$\sum KL_w(\beta_w||\phi_{wt}) \to \min_{\Phi}, \qquad \sum KL_t(\alpha_t||\theta_{td}) \to \min_{\Theta},$$

что с заранее заданными коэффициентами α_0 и β_0 дает следующий регуляризатор:

$$R(\Phi,\Theta) = \beta_0 \sum_{t\in T}\sum_{w\in V} \tilde{\beta}_w \ln \phi_{wt} + \alpha_0 \sum_{d\in D}\sum_{t\in T} \tilde{\alpha}_t \ln \theta_{td} \to \max_{\Phi,\Theta}.$$

В результате для пересчета параметров в обобщенных формулах для М-шага 3.8, 3.9 справедливы соотношения:

$$\phi_{wt} \propto \hat{n}_{wt} + \beta_0\beta_w, \qquad \theta_{td} \propto \hat{n}_{td} + \alpha_0\alpha_t$$

Этот же результат может быть получен байесовским выводом, который лежит в основе латентного размещения Дирихле [96]. По этой причине в [93] сглаживающий регуляризатор также называется регуляризатором Дирихле.

Регуляризация Дирихле приводит к сглаживанию частотных оценок ϕ_{wt} и θ_{td}, что является единственным принципиальным отличием LDA от классического PLSA. В [97] приводятся результаты экспериментов, согласно которым оптимальные значения гиперпараметров α_t и β_w оказываются близки к нулю. Оценки параметров ϕ_{wt} и θ_{td} в исходной модели вероятностного латентно-семантического анализа и латентного размещения Дирихле заметно отличаются только для терминов, редких в теме, и тем, редких в документе, которые не несут статистически значимой информации о тематике коллекции.

Таким образом, LDA оказывается лишь одной из множества моделей, которые можно получить в рамках вероятностного подхода к латентно-семантическому анализу с аддитивной регуляризацией. В качестве базовой модели логичнее брать более простую модель PLSA, которая не имеет собственных регуляризаторов, и добавлять к ней регуляризаторы, адекватные конкретной задаче.

3.4.3 Разреживание тематической модели

Выше уже упоминалось о том, что согласно гипотезе разреженности, каждый документ и каждый термин принадлежат небольшому числу тем. Очевидно, что использование сглаживающего регуляризатора действует прямо противоположным образом.

С практической точки зрения предпочтительными являются модели с сильно разреженными матрицами Φ и Θ, в которых доля нулевых значений превышает 90%. Разреживания можно добиться использованием *энтропийного регуляризатора*.

Поскольку среди всех распределений максимальной энтропией обладает равномерное распределение, логично «отдалить» искомые распределения от равномерного, т.е. максимизировать KL-дивергенцию между равномерными распределениями β_w и α_t и распределениями ϕ_t и θ_d. Энтропийный регуляризатор представляет собой сумму дивергенций по всем темам t и всем документам d с коэффициентами регуляризации β и α:

$$R(\Phi,\Theta) = -\beta_0 \sum_{t\in T}\sum_{w\in V} \ln \beta_w \phi_{wt} - \alpha_0 \sum_{d\in D}\sum_{t\in T} \ln \alpha_t \theta_{td} \to \max_{\Phi,\Theta}.$$

Формулы для M-шага для модели с энтропийным регуляризатором:

$$\phi_{wt} \propto (\hat{n}_{wt} - \beta_0\beta_w)_+ \qquad\qquad \theta_{dt} \propto (\hat{n}_{td} - \alpha_0\alpha_t)_+$$

Наряду с разреживанием и сглаживанием важными параметрами являются также однородность и контрастность тем. Ниже будет рассмотрен регуляризатор, управляющий контрастностью (различностью) тем модели [95].

3.4.4 Повышение различности тем

Одним из требований, предъявляемых к модели является смысловое различие между полученными темами. Формализовать интуитивное представление о различности тем можно по-разному.

Разумной выбором для меры различности тем является ковариация:

$$R(\Phi,\Theta) = -\frac{\tau}{2} \sum_{s\in T/\{t\}}\sum_{s\in T} cov(\phi_t,\phi_s) \to \max_{\Phi}$$

Этот критерий не зависит от Θ, поэтому для θ_{td} формула M-шага не меняется. Формула для ϕ_{wt}, принимает вид

$$\phi_{wt} \propto (\hat{n}_{wt} - \tau\phi_{wt} \sum_{sT/\{t\}} \phi_{ws})_+$$

Из формулы видно, что условные вероятности $p(w|t)$ уменьшаются для тех слов $w \in V$, которые имеют большие значения вероятности $p(w|s)$ в других темах. В процессе итераций EM-алгоритма для каждого слова вероятности наиболее значимых тем приобретают всё большие значения, а вероятности менее значимых тем уменьшаются. При больших τ требование некоррелированности приводит к разреживанию матрицы Φ [93].

3.4.5 Шумовые и фоновые термины в тематическом моделировании

Одной из важных модификаций описанного выше базового подхода является введение понятия *шума* и *фона* [93]:

Фон — это общеупотребительные слова, в частности, стоп-слова, имеющие значимые вероятности во многих темах.

Шум — это термины, специфичные для конкретного документа, либо редкие термины, относящиеся к темам, слабо представленным в данной коллекции. Тематическая модель даёт слишком низкие значения вероятности $p(w|d)$ для таких терминов, то есть не способна объяснить их появление в документах коллекции.

Предполагается, что исключение фона и шума поможет улучшить качество модели, поскольку описание распределения таких слов в рамках тематической модели не имеет смысла.

Одним из способов получения шумовой и фоновой тем является регуляризация нескольких выбранных тем в коллекции. В экспериментах, описываемых далее, для получения фоновой темы используются:

1. Сглаживающий регуляризатор, приближающий распределение слов в фоновой теме к распределению слов в коллекции β_w;

2. Сглаживающий регуляризатор, приближающий распределение фоновой темы по документам к равномерному;

3. Декоррелирующий регуляризатор, понижающий вероятности предметных тематических слов в фоновой теме. Таким образом, в фоновой теме остаются только слова общей лексики.

В результате фоновая тема присутствует практически во всех документах и содержит слова общей лексики, стоп-слова и редкие слова, которые исключаются из предметных тем в результате разреживания [98].

Для получения шумовой темы используются:

1. Сглаживающий регуляризатор, приближающий распределение слов в шумовой теме к некоторому распределению β'_w, в котором вероятность слова w *обратно* пропорциональна частоте по коллекции;

2. Сглаживающий регуляризатор, приближающий распределение шумовой темы по документам к равномерному;

В результате в шумовой теме содержатся редкие предметные слова, которые могут случайным образом появиться в любом документе коллекции.

Данный подход к определению шумовой и фоновой тем используется в экспериментах ниже.

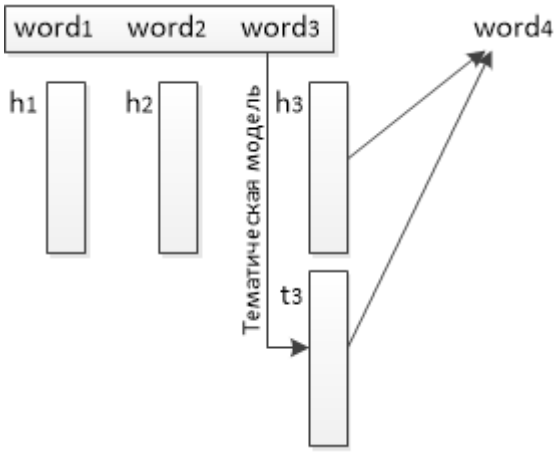

Рисунок 3.3: Схема работы предлагаемой модели. Рекуррентная нейронная сеть генерирует последовательность векторов скрытых состояний h_t, тематическая модель генерирует последовательность тематических разложений левого контекста t_3

3.4.6 Интерпретируемость тем

В [98] были предложены дополнительные меры интерпретируемости тематических моделей: *размер ядра темы*, *чистота темы* и *контрастность темы*.

Ядром темы $\mathbf{t}$ называется множество слов $W_{mathbft} : p(\mathbf{t}|w) > \delta$, $0 < \delta < 1$ — геперпараметр. *Размером ядра* $KS = |W_\mathbf{t}|$ называется соответственно количество элементов в $W_\mathbf{t}$.

Чистота темы $\mathbf{t}$ определяется как суммарная вероятность слов ядра:

$$P = \sum_{w \in W_\mathbf{t}} p(w|t)$$

Контрастность темы $\mathbf{t}$ равна средней вероятности встретить слова ядра именно в данной теме

$$C = \frac{1}{|W_\mathbf{t}|} \sum_{w \in W_\mathbf{t}} p(t|w)$$

Данные характеристики измерены в экспериментах ниже.

3.5 Гибридная языковая модель максимальной энтропии

Естественным способом добавления новой информации в рекуррентную нейронную сеть является расширение входного слоя за счет вектора тематического разложения левого контекста. К сожалению, добавление новой информации в рекуррентную нейронную сеть сопряжено с риском потери качества модели в результате попадания сети в локальные минимумы, соответствующие менее оптимальным показателям перплексии.

Альтернативой могло бы стать отдельное обучение рекуррентной модели с последующим объединением признаков, полученных из скрытого слоя сети на шаге t с тематическим профилем контекста на шаге t в рамках модели максимальной энтропии [20].

Фактически данный подход предполагает обучение простой многоклассовой логистической регрессии:

$$p(w_{t+1}|h_t, d_t) = \frac{e^{-v_w \cdot h_t - f_w \cdot d_t}}{\sum_{w'} e^{-v_{w'} \cdot h_t + f_{w'} \cdot d_t}},$$

где v_w — вектор весов слова w для элементов вектора скрытого состояния h_t, f_w — вектора весов слова w для вектора тематического разложения левого контекста d_t.

В предлагаемой модели стандартная рекуррентная сеть, обученная ранее на лемматизованном корпусе, используется для кластеризации левых контекстов (см. рис. 3.3): фактически на каждом шаге сеть выдает содержимое скрытого слоя. Полученное векторное представление контекста объединяется с вектором вероятностей $p(\mathbf{t}|d)$, полученных методами тематического моделирования. В результате будет получен вектор признаков, в котором будет отражена информация как о ближнем (скрытый слой рекуррентной сети), так и о дальнем контексте (распределение тем в документе).

Как и в случае с нейронной сетью, процедура обучения модели максимальной энтропии может быть весьма длительной, поскольку задействует большое количество матричных операций. По этой причине как рекуррентная нейронная сеть, так и модель максимальной энтропии были реализованы на GPU.

Поскольку тематическое разложение является достаточно дорогой процедурой, и можно предположить, что тематические свойства документа меняются достаточно медленно, тематическое разложение можно вычислять с некоторым заранее заданным шагом F_e в скользящем контекстном окне длины L. Оба гиперпараметра критически важны для качества модели и скорости вычислений. Как и в случае с обработкой речевых сигналов, выбор длины окна является компромиссом между точностью представления и изменчивостью свойств обрабатываемой последовательности. Выбор частоты вычисления тематического вектора в свою очередь критически влияет на производительность с одной стороны и на качество с другой.

3.6 Эксперименты

Тематическая модель была обучена на новостном корпусе Lenta.ru (апрель 2014 — март 2015).

Для экспериментов было обучено три тематических модели:

1. Модель с комбинацией сглаживающих, разреживающих и декоррелирующего регуляризаторов и дополнительными предположениями о шумовой и фоновой темах [98]. Далее будем обозначать ее как srPLSA.

2. Модель из предыдущего пункта, но без шумовой и фоновой тем. Далее будем обозначать ее как sPLSA.

3. Модель только со сглаживающим регуляризатором (аналог LDA). Далее будем обозначать ее как LDA.

Таблица 3.1: Характеристики тематических моделей

Модель	$\|W\|$	$Sp(\Phi)$, %	$Sp(\Theta)$, %	KS, %	P %	C, %	PPL
srPLSA	34725	71,9	98,1	123,3	95,1	70,1	410
sPLSA	34725	44,7	53,5	120,4	92,3	68,8	408
LDA	34725	0	0	66,2	47,1	52,1	393

Для всех тематических моделей измерялись следующие количественные характеристики: перплексия PPL, доля нулевых элементов матриц Φ и Θ ($Sp(\Phi)$ и $Sp(\Theta)$ соответственно), а также меры интерпретируемости тем: размер ядра KS (в словах), чистота P и контрастность C тем [98]. Количество тем во всех моделях было равно 300. Информация об используемых моделях сведена в таблицу 3.1.

Модель на рекуррентной нейронной сети и модель максимальной энтропии была натренирована на подкорпусе корпуса Lenta.ru объемом приблизительно в $2 \cdot 10^6$ токенов. Примерно 10% данных было выделено для валидации. Каждый текст был обработан морфологическим анализатором/лемматизатором для русского языка [74] со встроенным словарем примерно в $2 \cdot 10^6$ словоформ. Выходом анализат ора являлся текст, в котором все известные словоформы были заменены соответствующими леммами, а неизвестные — специальным токеном "UNK".

Обучение тематических моделей было реализовано на языке C++ с использованием библиотеки Eigen.

Обучение рекуррентной модели и модели максимальной энтропии было также реализовано на языке C++ с использованием расширений CUDA C и библиотеки CUBLAS для матричных вычислений на GPU. Обучение рекуррентной модели со скрытым слоем 500 и словарем (и, следовательно, выходным слоем) 15000 заняло около 20 часов на одной видеокарте NVIDIA GTX980 TITAN. Обучение модели максимальной энтропии занимает от 7 до 10 часов при той же конфигурации оборудования.

Базовая модель без применения тематического разложения достигает 284.28 по перплексии на тестовой выборке.

В первой серии экспериментов обучение модели максимальной энтропии производилось без предварительной нормализации данных. Предполагалось, что поскольку элементы скрытого слоя нейронной сети h и вероятности тем в тематическом разложении t лежат в диапазоне [0;1], нормализация не потребуется. Тем не менее, в результате экспериментов выяснилось, что нормализация исключительно важна: поскольку t является вероятностным распределением и выполняется вероятностное ограничение

$$\sum_i t_i = 1,$$

то средние значения элементов t оказываются оказываются гораздо ближе к 0, чем в случае вектора h. Таким образом, фактически информация о тематическом разложении документа оказалась проигнорирована: результат оказался независящим ни от, выбранной модели, ни от размера окна. Тем не менее, перплексия новой модели во всех случаях оказалась ниже исходной почти на 4%. Фактически исходная нейросетевая модель оказалась дообучена (*fine-tuned*) при фиксированной матрице рекуррентных весов W. Несмотря на то, что дообучение является стандартной практикой в использовании глубоких нейронных сетей, нам не известны описанные в статьях примеры его применения для языкового моделирования.

Во второй серии экспериментов обучение модели было запущено на выборке с нормализованными признаками. Нормализация стандартным отклонением:

$$x'_i = \frac{x_i - \bar{x_i}}{\sigma_{x_i}}$$

привело к неустойчивости в процедуре обучения, поэтому в качестве метода нормализации был выбран минимаксная нормализация:

$$x'_i = \frac{x_i - min(x_i)}{max(x_i) - min(x_i)}$$

Результаты экспериментов приводятся в таблице 3.2.

Таблица 3.2: Перплексия моделей на тестовой выборке.

Модель/размер скрытого слоя	**Перплексия**
RNN	284.28
RNN с дообучением	273.5
RNN+LDA 250	271.9
RNN+LDA 200	271.16
RNN+LDA 150	269.66
RNN+LDA 100	267.61
RNN+LDA 50	265.39
RNN+LDA 25	**265.38**
RNN+srPLSA 250	272.08
RNN+srPLSA 200	271.85
RNN+srPLSA 150	271.46
RNN+srPLSA 100	270.97
RNN+srPLSA 50	**270.13**
RNN+srPLSA 25	270.12
RNN+sPLSA 250	277.047
RNN+sPLSA 200	274.95
RNN+sPLSA 150	272.29
RNN+sPLSA 100	269.60
RNN+sPLSA 50	**266.68**
RNN+sPLSA 25	267.60

Таблица 3.3: Перплексии моделей с шагом вложения $F_e = 5$ и длиной скользящего окна $L = 25$

Модель	**Перплексия**
RNN+LDA	**254.23**
RNN+sPLSA	256

Из таблицы 3.2 и графиков на рис. 3.4 видно, что перплексия убывает с уменьшением длины окна. По-видимому, главной причиной данного явления можно считать характер корпуса: новостные статьи не отделены друг от друга и при этом не отсортированы по тематическим разделам. По этой причине на границах статей качество модели падает. Соответственно с уменьшением длины окна данный эффект уменьшается.

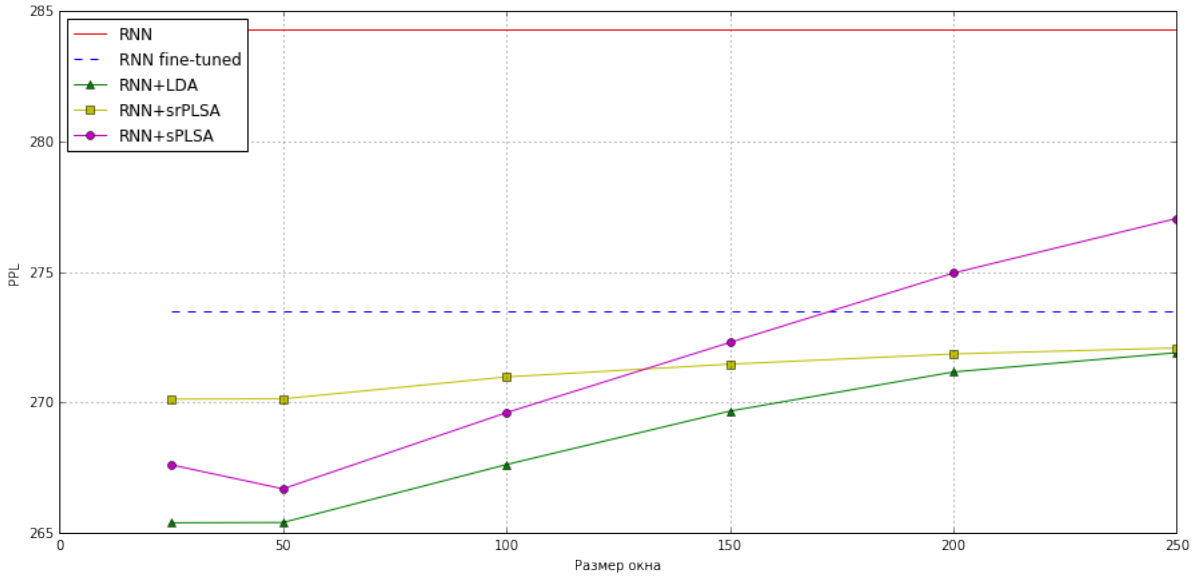

Рисунок 3.4: Зависимость качества языковой модели на основе тематического разложения от длины скользящего окна. Во всех случаях вложения вычислены с шагом $F_e = 25$.

Рассмотрим теперь каждый из графиков в отдельности. Худшей из моделей в смысле перплексии оказывается разреженная модель с шумом и фоном. Это объясняется тем фактом, что вероятности шумовой и фоновой тем, присутствующих в данной модели, остаются почти постоянными на всем корпусе и не несут никакой контекстной информации. Таким образом количество эффективных с точки зрения предсказания признаков уменьшается как минимум на 2. В модели без шума и фона этот недостаток устранен, однако она также уступает модели LDA, что указывает на то, что сглаживание, присутствующее в LDA-модели является существенным для предсказаний.

Для проверки влияния эффекта частоты вычисления тематического разложения был поставлен один эксперимент, результаты которого указывают на то, что данный гиперпараметр является критическим для качества конечной модели. В эксперименте сравнивались модель, основанная на LDA и разреженная модель без шума и фона. В обоих случаях шаг вложения $F_e = 5$, длина окна $L = 25$.

Из таблицы видно, что наилучший результат, достигнутый в экспериментах с длиной окна $L = 25$ был улучшен на 4.2%. Улучшение относительно исходной модели составило 10.56%, относительно дообученной модели — соответственно 7.04%. можно предложить как минимум две причины улучшения качества относительно модели с большим шагом вложения. Первая из них состоит в том, что тема, действительно ,меняется достаточно быстро и вычисление необходимо производить чаще. Вторая причина может состоять в том, что уменьшая шаг вложения, мы тем самым увеличиваем количество различных тематических векторов в выборке: по сравнению с моделью с $F_e = 25$ их становится в 5 раз больше, что приводит к лучшей обобщающей способности модели.

3.7 Выводы

В данной главе была предложена и проанализирована гибридная модель, основанная на рекуррентной нейронной сети и вероятностном тематическом моделировании. Предложенная модель фактически является моделью максимальной энтропии, на вход которой в качестве признаков поступают соответственно скрытый слой рекуррентной сети для моделирования языка и распределение тем в документе, вычисляемое методами тематического моделирования. Использование тематического моделирования мотивировано тем, что данный метод позволяет моделировать длинные семантические связи между словами и тем самым обойти проблему угасания градиента в рекуррентной сети.

В ходе экспериментов были выявлены следующие факты:

1. Описанная выше гибридная модель оказывается эффективной: в целом обучение модели ведет к более чем 10%-му снижению перплексии относительно исходной рекурретной модели.

2. Частично данное улучшение объясняется дообучением модели. Несмотря на то, что дообучение является стандартной практикой в использовании глубоких нейронных сетей, нам не известны описанные в статьях примеры его применения для языкового моделирования.

3. Наиболее эффективными для языкового моделирования являются признаки, полученные на основе модели LDA. Тем не менее, вероятностная тематическая модель с разреживанием позволяет добиться почти такого же уровня перплексии. Использование резреженности может быть важным фактором для хранения моделей большого размера.

4. В целом, качество гибридной модели растет с уменьшением длины скользящего окна и шага вычисления тематического разложения. Оптимальной кажется длина окна в 25–50 слов и минимально возможный шаг вычисления разложения.

Глава 4

Предсказание морфологических характеристик с помощью нейронной сети

4.1 Введение

В предыдущих главах речь шла о применении стандартной языковой модели на рекуррентной нейронной сети к русскому языку и обсуждались методы борьбы с проблемой угасания градиента. При этом все эксперименты выполнялись на лемматизованном корпусе. В данной главе речь пойдет о предсказании морфологических характеристик леммы. Поскольку предсказание морфологических характеристик является более простой задачей в смысле перплексии, а зависимости между элементами цепочки более короткими, использование рекуррентной архитектуры на данном этапе не является необходимым. В то же время полезное наблюдение о более тесной зависимости между соседними элементами указывает на полезность использования сверточных слоев.

Глава структурирована следующим образом. В разделе 2 будет рассмотрена модель с одновременным предсказанием леммы и морфологической формы и приведены результаты экспериментов. В разделе 3 описывается архитектура сверточной нейронной сети и ее применение к задаче предсказания морфологической характеристики. В разделе 4 приводятся результаты экспериментов. В разделе 5 описано применение полученных моделей к задаче распознавания речи. Наконец, в разделе 6 делаются основные выводы.

4.2 Языковая модель на рекуррентной нейронной сети с предсказанием морфологических признаков

Прямолинейным решением проблемы разреженности данных и размера выходного слоя нейронной сети является модификация исходной архитектуры, в которой выходной слой

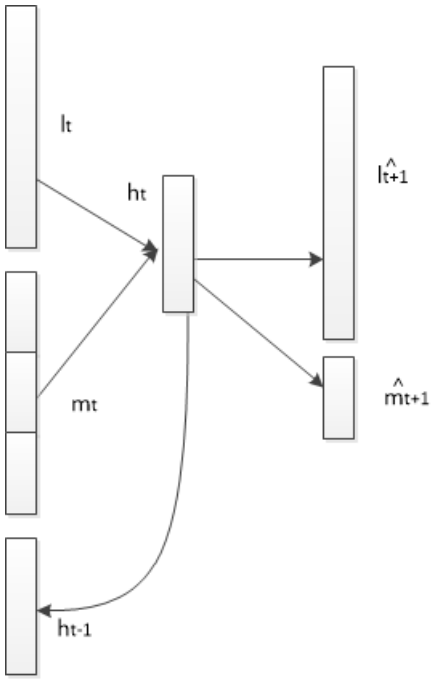

Рисунок 4.1: Рекуррентная нейронная сеть с двумя выходными слоями: распределение лемм $\hat{l}_{t+1}$ и морфологических признаков $\hat{m}_{t+1}$

расщепляется на два: слой, вычисляющий распределения лемм и слой-распределение морфологических признаков (см. рис. 4.1).

Несмотря на то, что интуитивно данный подход кажется привлекательным, эксперимент, приведенный ниже показывает, что ожидаемого результата он не приносит, по крайней мере, без дополнительных модификаций.

В эксперименте ниже проводилось сравнение стандартной архитектуры Т.Миколова с ее небольшими модификациями. Всего было рассмотрено три архитектуры.

Архитектура первого типа представляла собой стандартную архитектуру Т.Миколова (здесь и далее l_t, m_t — соответственно лемма и список морфологических признаков словоформы виде tag1@tag2@...@tagN на шаге t): леммы левого контекста предсказывали последующие леммы: $P(l_t|l_{t-1},h_{t-1})$.

Архитектура второго типа отличалась только наличием дополнительного вектора морфологических признаков. Таким образом, леммы и морфологические признаки левого контекста предсказывали последующие леммы: $P(l_t|l_{t-1},m_{t-1},h_{t-1})$.

Наконец, архитектура третьего типа осуществляла также предсказание морфологической формы: $\mathsf{P}(l_t,m_t|l_{t-1},m_{t-1},h_{t-1})$.

В последнем случае целевая функция представляет собой сумму целевых функций для леммы и класса морфологической формы:

$$\mathcal{L}(\theta) = \sum_{1\le t\le T} \mathcal{L}_t^l(\theta) + \sum_{1\le t\le T} \mathcal{L}_t^m(\theta)$$

Верхние индексы l и m обозначают целевые функции по леммам и морфологическим классам соответственно.

Скрытый слой h_t в случае второй и третьей модификаций вычисляется с учетом морфологических признаков предыдущей леммы:

$$h_t = f(W \cdot h_{t-1} + U_l \cdot x_t + U_m \cdot m_t + b)$$

U_l, U_m — матрицы вложений для лемм и морфологии соответственно; x_t, m_t — входные лемма и ее морфологические характеристики.

В рамках архитектуры третьего типа предсказания морфологических характеристик представляло собой классификацию на 200 классов, где каждый класс представлял собой полный набор морфологических характеристик. Например: N@МУЖ@ЕД@РОД@ОДУШ. Очевидным недостатком такого подхода является то, что предсказания морфологической формы делаются независимо от леммы, что вообще говоря делает возможным предсказания вида "гулять:N@МУЖ@ЕД@РОД@ОДУШ— то есть приписывание лемме невозможной морфологической характеристики. Это должно было явным образом отрицательно сказаться на перплексии. Эксперимент, однако, показал, что это не единственная и, видимо, не самая серьезная проблема данного подхода.

Для эксперимента был подготовлен новостной корпус, основанный на заметках издания Lenta.ru за март-ноябрь 2014 года. Корпус насчитывал $1.8 \cdot 10^6$ словоупотреблений. Корпус был обработан морфологическим анализатором [74] и преобразован в формат последовательностей вида лемма1:морфология1 лемма2:морфология2... леммаN:морфологияN. Словарь был ограничен 10000 лемм. Остальные леммы получали метку "UNK". По 10% корпуса были выделены для тестовой и валидационной выборок. Каждая из архитектур обучалась на компьютере, снабженном CUDA-совместимой видеокартой NVIDIA GTX TITAN. Были протестированы различные объемы скрытого слоя от 100 до 1000. Обучение каждой из сетей занимало около 20 часов почти вне зависимости от размера скрытого слоя, что является существенным улучшением по сравнению с используемой в [72] утилитой Т.Миколова [73], не использующей параллельных матричных вычислений.

Ниже приведены результаты работы алгоритмов на валидационной выборке. Перплексии на тестовой и валидационной выборке обычно разнятся в пределах 3–4 пунктов.

На рис. 4.2 lem2lem — базовая архитектура: на вход сети поступают леммы, на выходе генерируется распределение лемм; lem-morph2lem — на вход поступает дополнительный вектор морфологических признаков; lem-morph2lem-morph — на выходе дополнительно генерируется распределение морфологических форм.

Из таблицы 4.1 и графиков на рис. 4.2 видно, что перплексии моделей, использующих морфологию, всегда выше, чем для модели без морфологии. Если для модели номер 3 это можно было бы объяснить спецификой функции ошибки — фактически над одним и тем же множеством переменных строится два вероятностных распределения вместо одного, —

Таблица 4.1: Перплексии моделей на валидационной выборке

Модель/ Скрытый слой	**Леммы → Леммы**	**Леммы+Морфология→ Леммы**	**Леммы+Морфология→ Леммы+Морфология**
100	298.58	317.78	332.68/20.922
200	290.42	302.96	317.36/19.16
300	286.80	**296.81**	317.36/18.49
400	286.82	321.50	303.80/18.83
500	**286.22**	297.35	**302.75/18.95**
600	289.62	297.40	310.19/18.48
700	290.42	328.26	304.64/18.62
800	295.49	301.58	304.86/18.64
900	289.41	-	-
1000	291.23	-	-

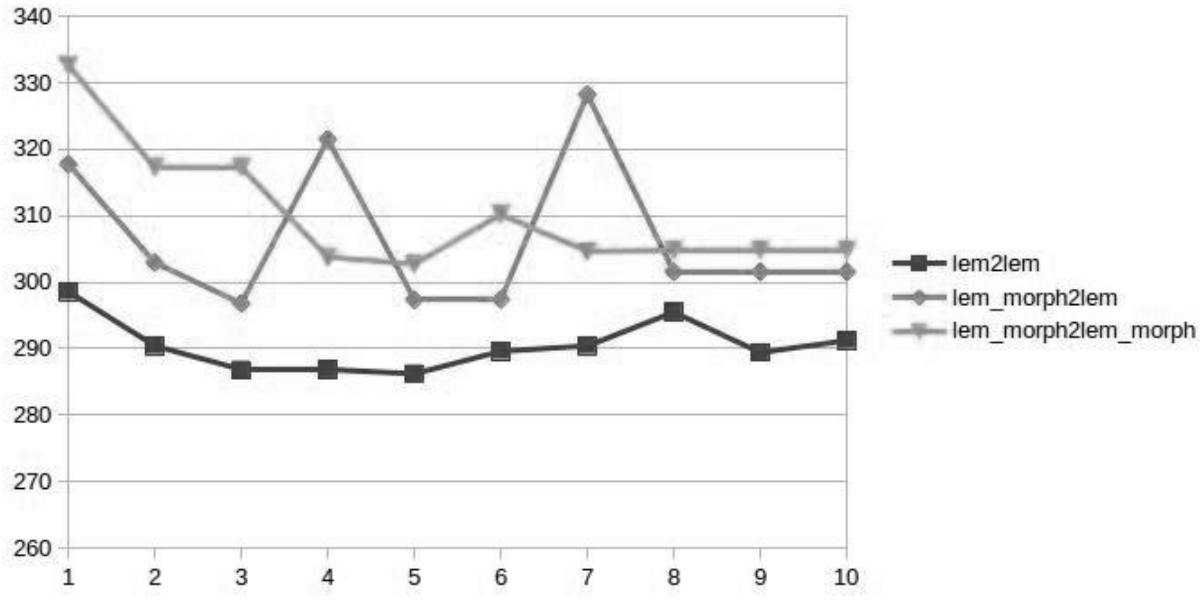

Рисунок 4.2: Зависимость перплексии по леммам от размера скрытого слоя для различных архитектур. Одно деление на оси абсцисс соответствует 100 элементам на скрытом слое

то падение качества для второй модели выглядит неожиданным. Одной из причин понижения качества могло бы стать переобучение вследствие увеличения числа параметров, однако различие в числе параметров нельзя назвать значительным ввиду небольшого числа морфологических классов в сравнении с количеством лемм.

Вполне вероятно, что более тщательный подбор гиперпараметров — количество нейронов на скрытом слое и регуляризация — позволили бы получить значительно лучшее качество. Подбор гиперпараметров является крайне затратной по времени операцией, поскольку количество итераций алгоритма обратного распространения ошибки, необходимых для оценки качества модели с данной конфигурацией гиперпараметров, как правило, довольно велико. При этом по результатам экспериментов нельзя выявить какого-либо тренда в зависимости перплексии на валидационной выборке от размера скрытого слоя. По крайней мере, нельзя с уверенностью утверждать, что увеличение размера скрытого слоя даст положительный эффект.

Самым разочаровывающим результатом является то, что перплексия, посчитанная для модели с предсказанием морфологических характеристик оказывается выше, чем для 5-граммной модели Кнесера-Нея.

Таким образом, по результатам поставленного эксперимента можно сделать вывод, что добавление дополнительных признаков в рекуррентную нейронную сеть является самостоятельной задачей, требующей исследования. Для предсказания лемм модель, игнорирующая морфологию, в целом представляется более надежной. С другой стороны, необходимо найти решение для задачи предсказания морфологических характеристик.

4.3 Предсказание морфологических характеристик леммы

Принимая во внимание результаты, изложенные в предыдущем разделе, можно сформулировать дальнейшую задачу как предсказание морфологических признаков леммы по известному левому контексту: $\mathsf{P}(m_t|l_1 \dots l_t, m_1 \dots m_{t-1})$, где лемма l_t предсказывается рекуррентной нейронной сетью, описанной в главе 2. Другими словами, предлагается использовать модель с внешним классификатором морфологических признаков, описанном в главе 2.

Данный подход предоставляет полную свободу в выборе классификатора морфологии: от n-граммных моделей до лесов решающих деревьев и нейронных сетей. Важным отличием задачи определения морфологических признаков от предсказания леммы является не только большое различие в количестве классов (15000 для лемм и не более 200 для морфологических признаков), но и характер зависимостей в последовательности: если важное для предсказания тематическое слово может находиться потенциально даже в предыдущем предложении, то зависимость между морфологическими признаками слов в предложении действует на более коротких расстояниях. Это предположение оправдывает использование n-граммных

моделей и классификаторов, использующих признаки n-грамм. С другой стороны, отказ от рекуррентной архитектуры означает, что вся информация о взаимодействии морфологических признаков будет задана в явном виде с помощью сложных признаков. Наличие сложных признаков в свою очередь приводит к необходимости их отбора для предотвращения переобучения, а также собственно их дизайна и извлечения. Дизайн признаков в свою очередь является достаточно трудоемкой задачей, требующей принятия априорных предположений о том, какие из морфологических характеристик могут влиять друг на друга.

Использование глубоких нейронных сетей позволяет избавиться от проблемы дизайна признаков, хотя и создает риски переобучения. С другой стороны, учитывая успехи в использовании нейронных сетей в обработке естественного языка [100] и значительный для такой сравнительно простой задачи размер выборки, можно надеяться на успешное обучение классификатора.

Ниже будут рассмотрен новый подход к обработке естественного языка при помощи сверточных нейронных сетей.

4.3.1 Сверточные нейронные сети в обработке естественного языка

Сверточные нейронные сети на текущий момент являются наиболее успешной нейросетевой архитектурой, зарекомендовавшей себя в распознавании изображений [101] [102], задачах анализа видео [103], и акустическом моделировании в распознавании речи [104].

Основным отличием сверточной архитектуры от архитектуры с прямым распространением является то, что вход нейрона i k-го слоя представляет собой линейную комбинацию лишь ограниченного подмножества активаций нейронов $k-1$ слоя. Каждое из подмножеств представляет собой множество нейронов, соседних с i на слое $k-1$, если представить его в виде многослойной двумерной сетки. В этом случае можно говорить о том, что k-й слой представляет собой дискретную свертку $k-1$ слоя с фильтром, коэффициенты которого обучаются методом обратного распространения ошибки [105].

Важным для понимания специфики применения сверточных сетей, в том числе и в задачах обработки текста, является то, что замена полносвязных слоев сверточными существенно уменьшает количество параметров и является более выгодным с вычислительной точки зрения. Упор на локальность признаков кажется скорее искусственным ограничением: так, в победившей на соревновании ImageNet-2014 архитектуре GoogLeNet [102] входом k-го слоя (т.н. *Inception* слой) является конкатенация нескольких сверток разного размера.

Интересно, что подход предпринятый в [102] ранее появился в задаче обработки естественного языка. А именно в оценке тональности текста [106].

Ниже данный подход рассмотрен подробно.

Рассмотрим цепочку слов $w_1 \dots w_N$. Обозначим за $v(w_i)$ d-мерный вектор, однозначно соответствующий данному слову. Метод отображения $\mathbb{V} \to \mathbb{R}^N$ вообще говоря не важен.

N-грамме $w_1 \dots w_N$ можно поставить в соответствие матрицу размера $N \times d$:

$$S = \begin{bmatrix} — & v(w_1)^T & — \\ — & v(w_2)^T & — \\ & \vdots & \\ — & v(w_N)^T & — \end{bmatrix}$$

Тогда свертка с «фильтром» размера $r \times d$ задает для каждой r-граммы последовательности $w_1 \dots w_N$ отклик на n-грамму. Задав K фильтров, получим отображение всей последовательности в новое признаковое пространство согласно формуле:

$$c_{k,i} = f(\sum_{i-\lceil r/2 \rceil + 1 < m < i + \lfloor r/2 \rfloor} \sum_{1 \le j \le d} a_{ijk} v(w_i)_j),$$

где a_{ijk} — параметры модели, f — нелинейная функция.

Одной из важных операций, используемых в сверточных нейронных сетях является пространственное прореживание выхода фильтра (*max-pooling*). Помимо прореживаниия также используется усреднение (*average-pooling*) и ряд других, например, иерархическое прореживание [105] [100]. Функция max-pooling слоев состоит с одной стороны в понижении размерности слоя. С другой стороны, max-pooling играет в некотором смысле роль нелинейной функции активации.

В приложениях сверточных сетей к обработке естественного языка операция прореживания приобретает временну*ю* интерпретацию: поскольку выход единственного сверточного слоя имеет размерность $l \times 1 \times K$, $l \ge 1$, то max-pooling осуществляется по сути лишь по первому измерению полученного тензора, т.е. «вдоль» n-граммы для каждого фильтра высоты r (см. рис. 4.3). Длина выхода l является гиперпараметром модели. Для вычисления значений c_{ik} при $i < \lceil r/2 \rceil$ и $i > N - \lfloor r/2 \rfloor$ используется дополнение матрицы S нулевыми строками.

Подход основанный на извлечении из цепочки $w_1 \dots w_N$ признаков сразу нескольких r-грамм определяет специфику сверточной архитектуры для обработки текста: входной слой обрабатывается сразу несколькими фильтрами параллельно, после чего выходы всех фильтров конкатенируются.

Как было отмечено в [102], такой подход можно трактовать как введение дополнительного ограничения на разреженность, таким образом, что все ненулевые веса модели собраны в непрерывные блоки. Такой вид разреженности дает важные преимущества в скорости вычислений на GPU.

В основном описанная выше архитектура используется для задач, требующих обработки относительно длинных цепочек слов [106] [107], однако использование сверточной сети для предсказания морфологии также выглядит перспективным: с одной стороны, использование нейронной сети позволяет уйти от проблемы ручного извлечения сложных признаков,

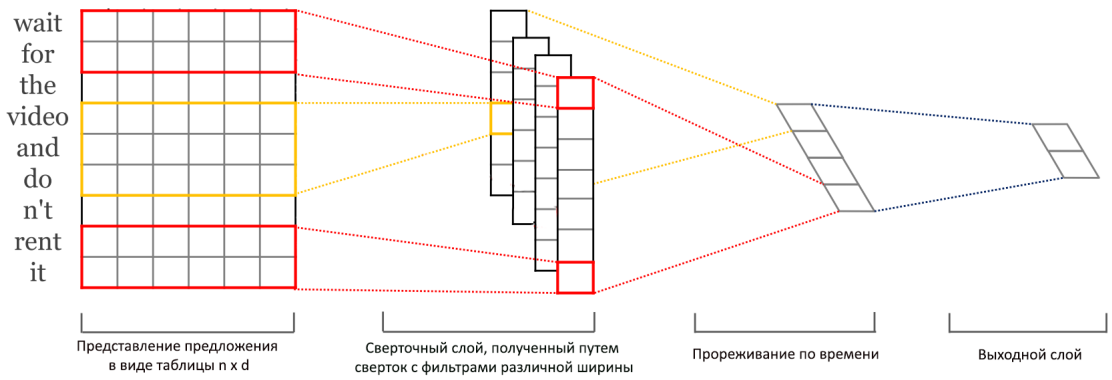

Рисунок 4.3: Сверточная нейронная сеть для обработки текста

с другой стороны, влияние контекста на морфологию осуществляется на существенно более коротких расстояниях между словами, так что использование достаточно длинных n-грамм, может быть предпочтительнее, чем, скажем, обучение рекуррентной модели. Преимуществом сверточной сети перед рекуррентной, безусловно, является скорость обработки и, что важнее, скорость обучения: обучение сверточной сети может осуществляться параллельно, т.е. обработка различных подмножеств обучающей выборки может проводиться независимо. В случае рекуррентной сети, в силу необходимости вычисления векторного вложения контекста, обработка выборки ведется последовательно, если не вводить дополнительных допущений.

Скорость эксперимента важна, поскольку делает возможным более точный подбор гиперпараметров, что в конечном счете влияет на качество модели. И это является весомым аргументом в сторону сверточной архитектуры.

4.4 Эксперименты по классификации морфологии

Ниже будут приведены результаты экспериментов с моделями предсказания морфологии на основе сверточной нейронной сети с прямым распространением.

Как и в предыдущих экспериментах, для обучения и тестирования использовался новостной корпус сайта Lenta.ru за 2014 год. Объем корпуса составил 3211256 токенов. Корпус был обработан морфологическим анализатором [74] и состоял из последовательностей пар вида «лемма:тег». Формат морфологического тега описан в главе 3.

Стоит отметить, что предлагаемая ниже модель опираются на предварительный морфологический анализ, а значит может быть использована только на этапе переранжирования гипотез и только в сочетании с языковой моделью на леммах. Часть речи и лемма предполагаются известными.

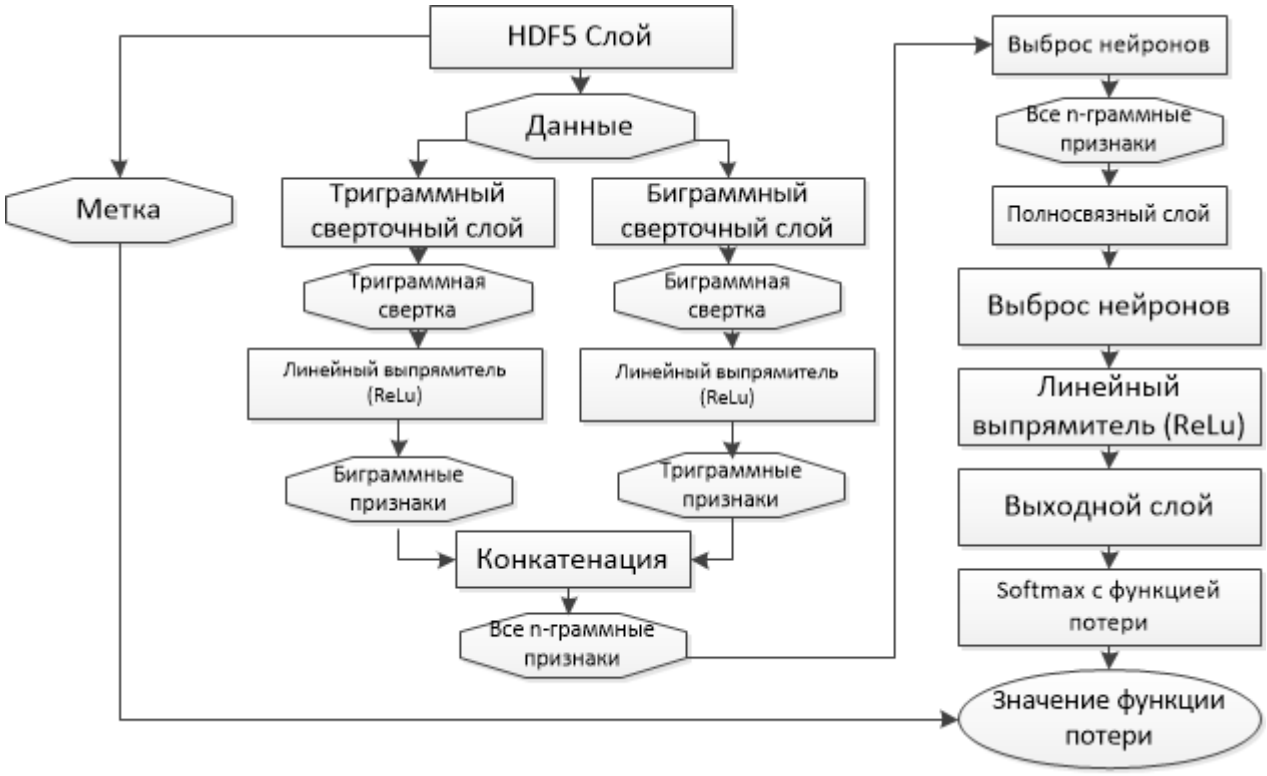

Рисунок 4.4: Схема сверточной нейронной сети для предсказания морфологии в библиотеке Caffe

4.4.1 Модель на сверточной нейронной сети

Модель предсказания морфологии с помощью сверточной нейронной сети опирается на архитектуру нейронной сети с одним сверточным и несколькими полносвязными слоями. Последний слой осуществляет классификацию на n классов, где n — количество возможных морфологических тегов для данной части речи.

Использование нейросетевого подхода к классификации морфологии оправдано в силу отсутствия необходимости ручного выбора сложных признаков: каждый элемент n-граммы на шаге t можно представить в виде вектора $m(k) \in \{0,1\}^{|G|}$, $t - n + 2 < k < t + 1$, где $\mathbb{G}$ — множество грамматических помет. Элемент $m_i(k) = 1$ тогда и только тогда, когда тег на шаге t содержит i-ю морфологическую метку.

Подход на основе одной морфологии был бы неполным, поскольку известно, что семантика также влияет на форму окружающих слов. По этой причине вектор $m(k)$ дополняется двумя векторами $v(k)$ и $v(t+1)$, полученными в word2vec-модели [80] [108]. $v(k)$ представляет собой векторное представление леммы на шаге k, вектор $v(t+1)$ — соответственно векторное представление известной леммы на шаге $t + 1$, морфологический тег которой необходимо предсказать. Таким образом размерность представления каждого элемента в n-грамме составляет $2d + |\mathbb{G}|$, где d — размерность векторного представления лемм.

Входом классификатора является матрица $S_{n\times(2d+|G|)}$, причем $\forall i S_{i,1:d} = v(t + 1)$. Данное решение не выглядит красивым, однако позволяет отразить в сверточной архитектуре факт взаимосвязи каждого элемента n-граммы с целевым словом.

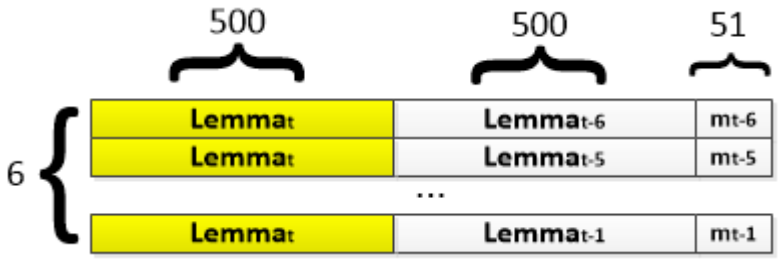

Рисунок 4.5: Формат входных данных сверточной нейронной сети для классификации морфологии

Для обучения и предсказания использовалась популярная библиотека для сверточных нейронных сетей Caffe, разработанная в Университете Беркли [109]. В отличие от многих других пакетов, Caffe полностью реализована на C++, что дает преимущества в производительности. Архитектура сети в Caffe задается через специальный формат prototxt, разработанный в Google. Использование prototxt позволяет получать из конфигурационных файлов компилируемый код на C++. Таким образом, каждая реализация нейронной сети полностью реализуется на C++ без использования отдельных скриптовых языков или парсера конфигурационных файлов.

В коде и документации Caffe принята несколько нетрадиционная трактовка понятия слоя. «Слоем» в Caffe называется любой модуль, преобразующий данные. Таким образом, функция, поточечно применяющая ReLU-функцию ко входному вектору, будет вызываться слоем ReLU. Так, на рис. 4.4 слои обозначены прямоугольниками, а данные — восьмиугольниками.

Данные читаются из базы в формате hdf5. Caffe поддерживает большое количество различных источников данных, однако в силу того, что Caffe предназначена в первую очередь для анализа изображений, как правило, подразумевается, что входной элемент выборки представляет собой массив чисел $0 - -255$. Hdf5 выгодно отличается тем, что может содержать числа с плавающей точкой. Поскольку, как было сказано выше, входные данные содержат word2vec-представления словоформ, использование hdf5 оказывается критичным. Векторные представления лемм вычислялись на основе модели из работы [108], обученной на корпусе библиотеки LibRuSec объемом 10^9 словоформ.

На этапе обучения из источника данных извлекается собственно элемент выборки и соответствующая метка. Элемент выборки представляет собой матрицу размерности $(500 + 500 + 51) \times 7$: в каждой i-й строке матрицы содержится конкатенация векторного вложения предсказываемой леммы и векторного вложения i-й леммы в 7-грамме, а также бинарного вектора из 51 морфологической пометки слова (см. рис. 4.5).

Далее элемент выборки параллельно обрабатывается двумя наборами фильтров — размерностей 2×1051 и 3×1051, соответствующих биграммным и триграммным признакам. Далее полученные представления конкатенируются в соответствующем слое, после чего следует два полносвязных слоя. Параметры слоев приведены в таблице 4.2.

Таблица 4.2: Список слоев нейронной сети для предсказания морфологии

Слой	
данные (1, 6, 1051)	
биграммная свертка (1024, 2, 1051)	триграммная свертка (1024, 3, 1051)
линейный выпрямитель (ReLU)	линейный выпрямитель (ReLU)
конкатенация	
выброс нейронов (dropout), p=0/5	
полносвязный слой(1024, 1000)	
выброс нейронов (dropout), p=0.5	
линейный выпрямитель (ReLU)	
полносвязный слой (1000, кол-во классов C)	

Таблица 4.3: Кросс-энтропия на тестовой выборке при предсказании точных форм для изменяемых частей речи

Часть речи (пометка)	**Количество классов**	**Кросс-энтропия**
Существительное (S)	113	0.718
Глагол (V)	118	0.938
Наречие (ADV)	3	0.037
Прилагательное (A)	32	1.067
Числительное (NUM)	48	1.025

Поскольку лемма и часть речи предполагаются известными, можно обучить отдельный классификатор для каждой части речи, что позволит полностью исключить возможность предсказания невозможных тегов для входной части речи. Очевидно, что в этой ситуации размер выходного слоя и количество классов различны, что сказывается на значении кросс-энтропии. Обучение единой модели для всех частей речи также возможно, но обучение отдельных моделей имеет два технических преимущества: 1) обучение отдельных моделей можно вести параллельно; 2) количество параметров в каждой из моделей будет меньше, а значит скорость обучения будет выше.

Результаты приведены в таблице 4.3.

В экспериментах использовалась версия Caffe ветки NVIDIA, запущенная на 1 GPU NVIDIA GTX Titan с поддержкой библиотеки CuDNN 3.

На рисунках 4.6–4.8 приведены графики обучения для существительных, глаголов и прилагательных. Во всех случаях тренировка прерывалась после 150000 итераций. Из графиков видно, что на данном этапе качество модели еще не выходит в насыщение и в дальнейших экспериментах результат может быть улучшен. Особенно хорошо это видно на примере существительных.

Несколько неожиданным выглядит результат модели для прилагательных: несмотря на меньшее, по сравнению с существительными, число классов, энтропия остается довольно высокой и убывает медленно с числом итераций. Возможно причина состоит в том, что как

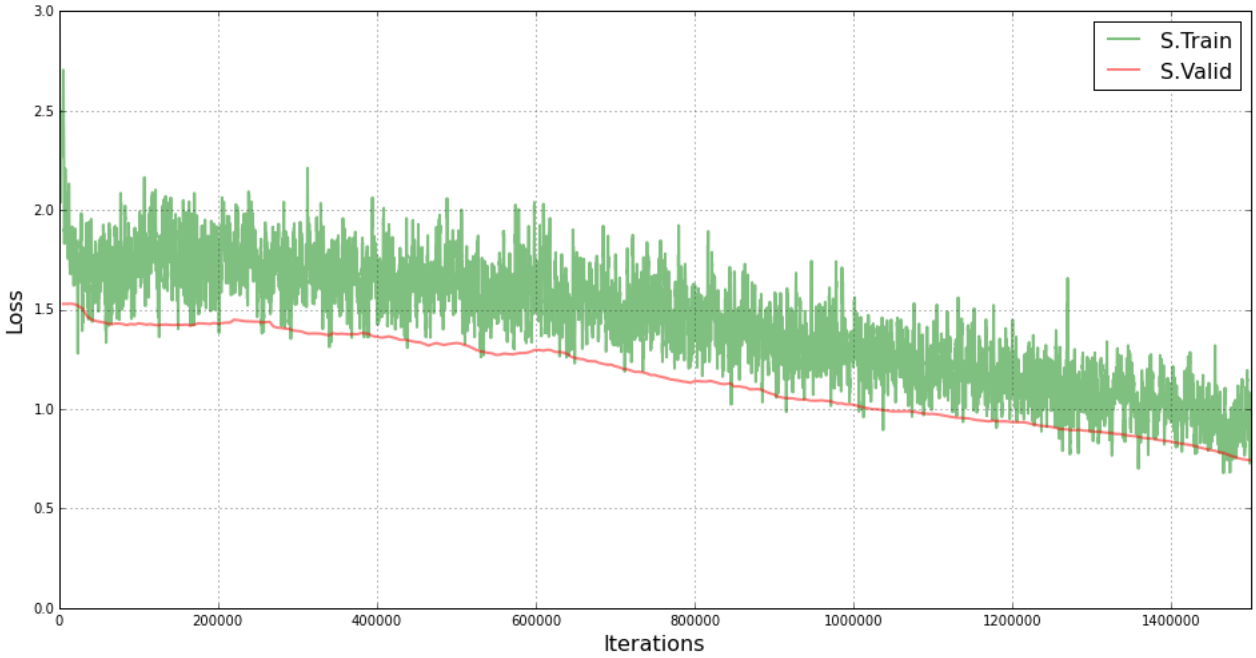

Рисунок 4.6: Зависимость кросс-энтропии на валидационной и тренировочной выборке от итерации в алгоритме обратного распространения ошибки. Модель для существительных

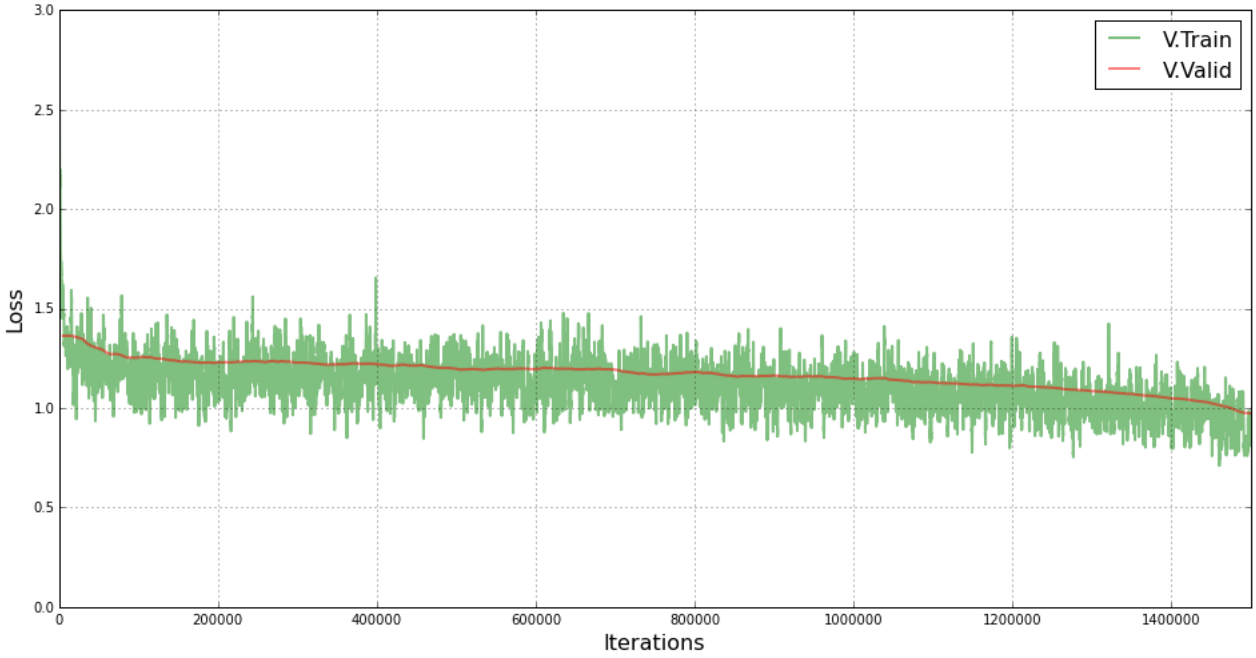

Рисунок 4.7: Зависимость кросс-энтропии на валидационной и тренировочной выборке от итерации в алгоритме обратного распространения ошибки. Модель для глаголов

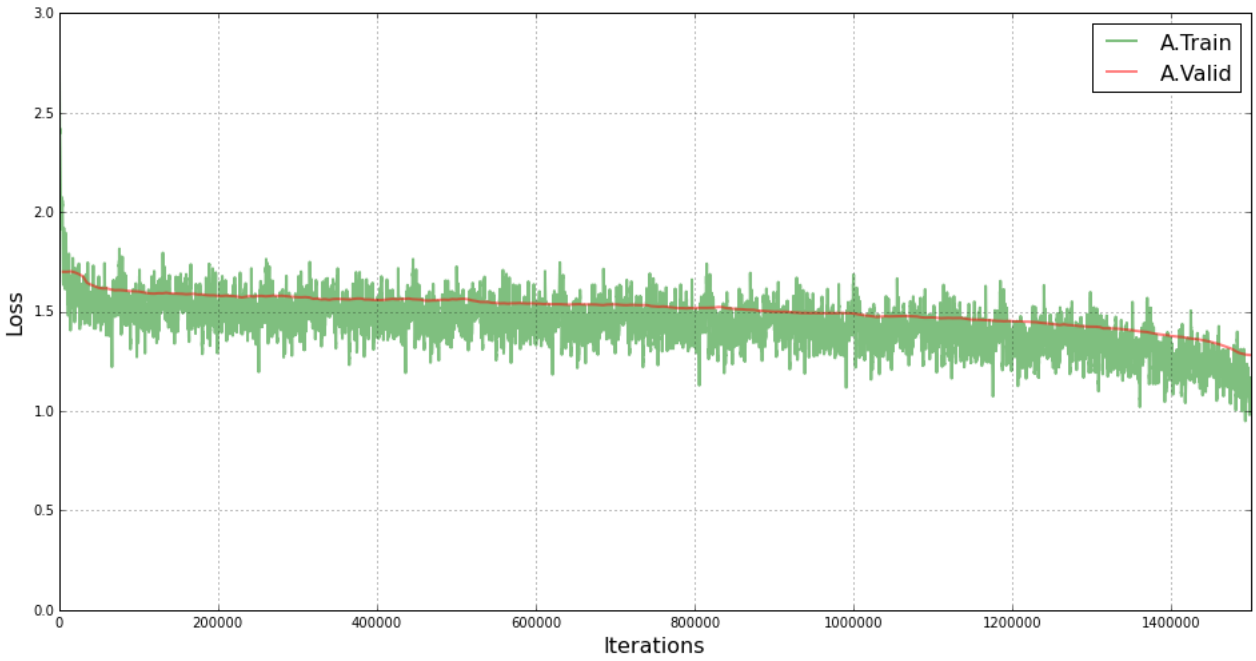

Рисунок 4.8: Зависимость кросс-энтропии на валидационной и тренировочной выборке от итерации в алгоритме обратного распространения ошибки. Модель для прилагательных

правило, прилагательные находятся в препозиции относительно определяемого слова. Таким образом, признаки, по которому можно было бы предсказать форму прилагательного по левому контексту сильно ограничены. С другой стороны, при известных признаках прилагательного, признаки существительного могут быть предсказаны достаточно хорошо. Этим, вероятно, объясняется столь сильное различие в результатах для моделей именных частей речи.

Для модели существительных, с самым большим объемом обучающих данных, тренировка заняла порядка 16 часов.

4.5 Эксперименты по распознаванию речи

Большая часть поставленных ранее экспериментов основывались на оценке перплексии исследуемых моделей. Ниже будет описан эксперимент, поставленный с использованием реальной системы распознавания речи для русского языка, основанной на известной библиотеке Kaldi.

4.5.1 Библиотека Kaldi

Kaldi — свободная библиотека для распознавания речи, полностью реализованная на языке C++. Управляющие сценарии реализованы на скриптовых языках — bash и Perl [110].

Особенностью Kaldi по сравнению с другими библиотеками является широкое использование аппарата взвешенных конечных преобразователей (ВКП, weighted finite-state transducers, WFST). Процесс распознавания представляется в виде поиска пути с минимальным весом в

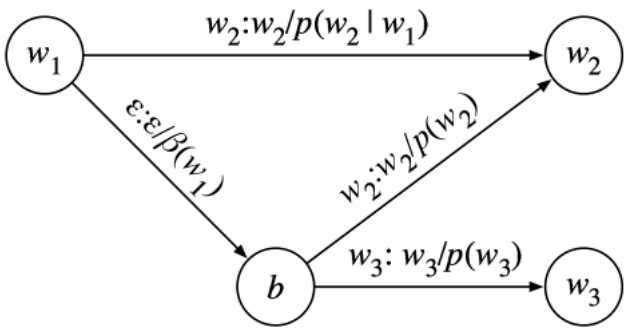

Рисунок 4.9: Backoff модель в Kaldi [112]

конечном преобразователе, отображающем последовательность состояний акустической модели (например, модели Бакиса [9]) на последовательность слов [111]. Согласно данной модели распознаватель может быть получен *композицией* преобразователей, соответствующих уровням модели: от акустической до языковой.

Другой особенностью Kaldi является включение в дистрибутив готовых сценариев для работы с популярными корпусами звучащей речи, — например, корпус Wall Street Journal, распространяемый Linguistic Data Consortium.

Идеология Kaldi предполагает реализацию на C++ только базовых алгоритмов. Потоки данных должны быть реализованы в скриптах на bash, Perl или Python. В настоящий момент Kaldi является одной из самых динамично развивающихся библиотек для распознавания речи.

4.5.2 Языковые модели в Kaldi

Простейшие языковые модели — заданные в виде грамматик или обученные на корпусе n-граммные модели — могут быть легко представлены в виде взвешенного конечного автомата. Например, биграммная модель может быть представлена в виде автомата где, каждое состояние соответствует слову, а веса на дугах соответствуют вероятностям соответствующих биграмм. Для триграммной модели состояния будут соответствовать уже парам слов.

Реализация сглаженных n-граммных моделей в Kaldi основана на добавлении в преобразователь специальных состояний, переход в которые не сопровождается выводом символа, а вес соответствует backoff-вероятности для соответствующей (n-1)-граммы [111].

Пусть, например, биграмма w_1w_2 встретилась в обучающем корпусе достаточное число раз, а биграмма w_1w_3 нет. Тогда в автомате G, задающем биграммную модель, вес перехода из состояния с меткой w_1 в состояние с меткой w_2 будет равен $-\log p(w_2|w_1)$. В то же время переход из w_1 в w_3 будет совершаться через дополнительное состояние $\beta(w_1)$, и полный вес пути будет равен $-\log(\beta(w_1)p(w_3))$. Фрагмент преобразователя, иллюстрирующий данных подход приведен на рис. 4.9.

Из рисунка видно, что при таком подходе в преобразователе присутствует уже два пути, ведущих в состояние w_2. В этом смысле автомат не соответствует исходной backoff-модели, а

лишь приближает ее в том смысле, что для большинства корректно оцененных n-граммных моделей путь в преобразователе и, следовательно, его вес будет совпадать с представлением вероятности $p(w_1 w_3)$ в модели, поскольку путь через backoff узел окажется менее оптимальным, чем непосредственный переход [112].

Для использования более сложных языковых моделей, не сводимых к аппарату конечных преобразователей, в Kaldi предусмотрен стандартный для системы автоматического распознавания речи подход: выдача взвешенных гипотез. Таким образом, языковые модели, основанные на рекуррентных нейронных сетях, могут быть применены после первого прохода, осуществляемого взвешенным конечным преобразователем, основанным на сглаженной n-граммной модели. Результатом первого прохода является список гипотез с двумя весами — весом, присвоенным акустической моделью, и весом, присвоенным сглаженной n-граммной языковой моделью. Первичный список отсортирован по сумме этих весов. Оценив веса, присвоенные нейросетевыми моделями, можно изменить сортировку списков с целью улучшения доли правильно распознанных слов, и, следовательно, качества системы распознавания речи.

4.5.3 Экспериментальная выборка

Для экспериментов по распознаванию речи использовался речевой корпус на русском языке [113]. Ниже описан состав корпуса и процесс его обработки для адаптации к формату, совместимому с Kaldi. Речевой корпус состоит из изолированных высказываний (предложений), которые можно разделить на следующие группы:

– фонетически полное множество, состоящее из 70 предложений и предназначенное для обучения;

– множество, состоящее из 3060 предложений, также предназначенное для обучения;

– множество, состоящее из 1000 предложений, составляющее тестовую выборку;

– множество, состоящее из 1000 предложений и предназначенное для отладки и оптимизации гиперпараметров (development set).

Первая из перечисленных групп была подобрана таким образом, чтобы обеспечить её фонетическую полноту. Группа содержит все допустимые фонемы из разработанной для корпуса фонетической системы (базируется на системе Аванесова), причем каждая фонема встречается в этом множестве не менее трех раз.

Для формирования остальных групп использовались тексты из различных газетных статей таких газет, как

– «Известия»,

– «Аргументы и факты»,

– «Демократический выбор России»,

– «Московский комсомолец»

и некоторых других, а также тексты из новостных сайтов в Интернете до 2001 года. Статьи подбирались таким образом, чтобы их содержание относилось к различным областям политики, экономики, культуры, искусства, медицины, спорта и т.д. При этом некоторые предложения брались целиком, без изменений, а другие несколько корректировались, с тем, чтобы они содержали, как правило, не более 9–10 слов.

4.5.4 Состав дикторов

Для произнесения предложений было привлечено 237 дикторов в возрасте от 18 до 65 лет, в том числе 127 мужчин и 110 женщин. Все они не являлись профессиональными дикторами и не имели опыта в искусстве речевого чтения. Распределение дикторов по возрасту:

– 18–20 лет—56 дикторов;

– 21–30 лет—101 диктор;

– 31–40 лет—33 диктора;

– 41–50 лет—25 дикторов;

– 51–60 лет—16 дикторов;

– 61–65 лет—6 дикторов.

Распределение дикторов по регионам:

– Москва—158;

– Подмосковье—2;

– Санкт-Петербург—3;

– Центральный регион—10;

– Среднее Поволжье—8;

– Нижнее Поволжье—3;

– Северный регион—4;

– Южный регион и Северный Кавказ—9;

– Татарстан—3;

– Урал—8;

– Сибирь—5;

– Дальний Восток—2;

– Украина—7;

– Крым— 2;

– Белоруссия—1;

– Молдавия—1;

– Грузия—2;

– Казахстан —2;

– Средняя Азия—5;

– Дальнее зарубежье —2.

Название каждого из файлов корпуса является кодом, из которого можно получить код диктора, соответственно разделив файлы по дикторам, основываясь на имени файла. Данная возможность является важной при тренировке акустической модели в Kaldi.

4.5.5 Формат звуковых файлов

Корпус содержал звуковые файлы в формате Microsoft Windows RIFF (Resource Interchange File Format) WAV с частотой дискретизации 44кГц, стерео, м/ф Plantronics.

Каждый файл снабжался файлом аннотации, содержащим информацию о дикторе, графемную и фонетическую транскрипции в каноническом и актуальном вариантах.

4.5.6 Подготовка данных

Стандартный сценарий работы с Kaldi предполагает помимо всего прочего конструирование словаря и n-граммной языковой модели. Для обеспечения приемлемого качества распознавания необходимо использовать словарь и корпус значительного объема.

Конструирование словаря и обучение языковой модели осуществлялось в несколько этапов:

– Извлечение первичного словаря и транскрипций из файлов аннотаций речевого корпуса;

– Тренировка языковой модели и извлечение словника;

– Обработка словника и его объединение с первичным словарем.

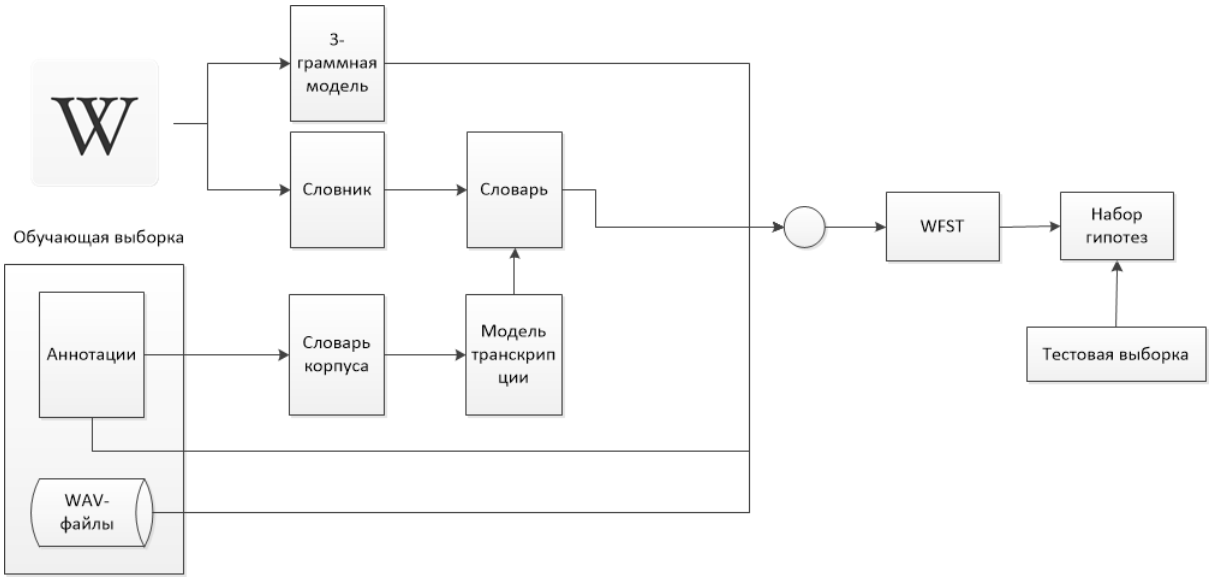

Рисунок 4.10: Схема процесса подготовки данных

Для получения первичного словаря была написана небольшая программа на языке Python, осуществлявшая выравнивание слов в графемной и фонемной записях файла аннотации, а также упрощение транскрипции: игнорировались контекстные изменения гласных, такие как упереднение.

В качестве n-грамной модели для Kaldi была выбрана 3-граммная модель со сглаживанием Кнесера-Нея, обученная на подмножестве статей архива русского раздела сайта «Википедия». Объем обучающего корпуса составил около 20 млн. словоформ. Тренировка модели осуществлялась с помощью пакета SRILM [114]. Объем словника составил порядка 100 тыс. словоформ.

Для получения транскрибированного словаря из словника была использована специально написанная утилита для расстановки ударений и автоматический транскриптор, основанный на скрытых марковских моделях, описанный в статье [115].

Для тренировки модели транскриптора был использован первичный словарь, в котором была выполнена расстановка ударений (в графемной записи) на основе фонемной записи.

Общая схема подготовки данных приведена на рис. 4.10.

Файл языковой модели в формате *arpa*, словарь, звуковые файлы и транскрипции являлись входом для специально подготовленных скриптов на Python и bash, преобразовывавших данные во входной формат Kaldi.

Далее производилось конструирование распознающего ВКП и тренировка акустической модели. Наилучшая модель с трифонами показала WER 17.8%.

Из тестовых данных генерировался список гипотез (10 на каждый файл), который далее обрабатывался морфологическим анализатором, использовавшимся ранее в главах 2 и 3. Полученные файлы подавались на вход языковым моделям: лексической модели на рекурентной нейронной сети и модели для предсказания морфологии.

Таблица 4.4: Результаты эксперимента по переранжированию гипотез. CNN — сверточная нейронная сеть для предсказания морфологии. λ — коэффициент при рекуррентной и сверточной моделях в интерполяции, приводящий к наибольшему падению WER

Модель	λ	**WER**, %
3-gram	0	17.80
3-gram + RNNLM	0.01	17.79
3-gram + CNN	0.2	17.2

4.5.7 Результаты эксперимента

Результаты эксперимента по переранжрованию приведены в таблице 4.4.

Из таблицы видно, что рекуррентная модель на леммах не дает практически никакого улучшения. Данный факт можно объяснить значительным различием размеров обучающих выборок в случае 3-граммной и рекуррентной моделей: размер обучающего корпуса отличался в 10 раз. Несмотря на то, что для тренировки модели использовалась реализация на GPU, для обучения модели на корпусе сопоставимого размера по-прежнему требуется время на порядок превосходящее время тренировки сглаженной n-граммной модели. Необходимо отметить, что в эксперименте не были задействованы более поздние модификации модели, направленные на ускорение обучения, так как такие модификации приводят к снижению качества по сравнению с канонической моделью. Стоит, однако, признать, что возможное снижение качества компенсировалось бы большим размером обучающей выборки, поэтому в будущем такой эксперимент будет поставлен.

Использование сверточной модели для предсказания морфологии приводит к некоторому улучшению, однако из графиков обучения, приведенных выше, что результат может быть улучшен, поскольку кросс-энтропия продолжает убывать. Тем не менее, окончательный результат может быть получен только при наличии рекуррентной модели для предсказания лексики с размером словаря и объемом обучающей выборки, соответствующим размеру словаря и тематическому разнообразию речевого корпуса. Данный эксперимент будет поставлен в будущем. В случае получения положительного результата в эксперименте с использованием рекуррентной модели на леммах, станет возможным постановка эксперимента с моделью, использующей тематическое моделирование.

4.6 Результаты и выводы

В данной главе была рассмотрена задача предсказания морфологических характеристик леммы с помощью нейронной сети. Решение данной задачи позволило бы снизить размер выходного слоя рекуррентной нейронной сети, ограничив его только леммами.

В главе были получены следующие результаты:

– Экспериментально установлено, что разбиение выходного слоя на слой лемм и морфологических классов с суммированием сигнала ошибки и сохранением одного скрытого

слоя приводит к снижению качества модели по сравнению с моделью, использующей только леммы.

– Эксперимент показал, что добавление морфологических характеристик на вход сети, также не приводит к улучшению предсказания лемм. Отрицательный результат можно объяснить необходимостью усложнения архитектуры сети (например, использования dropout-регуляризации) или более тщательным подбором гиперпарамтеров, что затруднено ввиду временных затрат на обучение модели.

– Была обучена модель предсказания морфологии с помощью сверточной нейронной сети. При сохранении архитектуры сети для каждой изменяемой части речи русского языка была натренирована отдельная модель.

– Рекуррентная модель для предсказания лемм и сверточная модель для предсказания морфологии были использованы для ранжирования гипотез, возвращаемых системой распознавания Kaldi. Для этого были натренированы акустическая модель русского языка и 3-граммная модель со сглаживанием Кнесера-Нея.

– Результаты эксперимента говорят о том, что рекуррентная модель на леммах, обученная на новостном корпусе в 2 млн. словоупотреблений не приводит к улучшению качества распознавания в комбинации с 3-граммной моделью со сглаживанием Кнесера-Нея, обученной на корпусе в 20 млн. словоупотреблений. Большой объем обучающего корпуса в последнем случае был необходим для обеспечения приемлемого качества первичных гипотез распознавания. Таким образом, необходима рекуррентная модель с большим размером словаря, обученная на корпусе большего размера и с более широким охватом тем, чем у новостного корпуса.

– Использование сверточной сети для предсказания морфологии ведет к снижению уровня пословной ошибки с 17.8% до 17.1%, причем есть основания считать, что модель может быть далее дообучена

– Необходимо проведение дальнейших экспериментов с комбинированием рекуррентной модели на леммах, тематическим моделированием и моделями морфологии.

Заключение

Основные результаты работы заключаются в следующем.

Предложена модель, предполагающая раздельное предсказание лемм и морфологии на основе нейронных сетей с целью уменьшения количества классов, соответствующих словоформам.

В рамках данного подхода была предложена и протестирована модель предсказания лемм на основе рекуррентной нейронной сети для русского языка. Эксперимент продемонстрировал, что проблема свободного порядка слов не является существенной для данной модели. Рекуррентная модель оказывается более эффективной, чем 5-граммная модель со сглаживанием Кнесера-Нея. Преимущество наблюдается как в эксперименте по измерению перплексии, так и в эксперименте по ранжированию гипотез распознавания при одинаковых обучающих данных.

Для улучшения базовой архитектуры рекуррентной нейронной сети, в частности, борьбы с нежелательным эффектом угасания градиента, была предложена и проанализирована гибридная модель, основанная на рекуррентной нейронной сети и вероятностном тематическом моделировании. Предложенная модель фактически является моделью максимальной энтропии, на вход которой в качестве признаков поступают соответственно скрытый слой рекуррентной сети для моделирования языка и распределение тем в документе, вычисляемое методами тематического моделирования. Использование тематического моделирования мотивировано тем, что данный метод позволяет моделировать длинные семантические связи между словами и тем самым обойти проблему угасания градиента в рекуррентной сети.

В ходе экспериментов с предложенной моделью было показано, что гибридная модель оказывается эффективной: в целом обучение модели ведет к более чем 10%-му снижению перплексии относительно исходной рекурретной модели. Частично данное улучшение объясняется дообучением модели. Несмотря на то, что дообучение является стандартной практикой в использовании глубоких нейронных сетей, нам не известны описанные в статьях примеры его применения для языкового моделирования.

Наиболее эффективными для языкового моделирования оказались признаки, полученные на основе модели LDA. Тем не менее, вероятностная тематическая модель с разреживанием позволяет добиться почти такого же уровня перплексии. Использование резреженности может быть важным фактором для хранения моделей большого размера.

В целом, качество гибридной модели растет с уменьшением длины скользящего окна и шага вычисления тематического разложения. Оптимальной кажется длина окна в 25–50 слов и минимально возможный шаг вычисления разложения.

Эксперименты с предсказанием морфологии показали, что разбиение выходного слоя на слой лемм и морфологических классов с суммированием сигнала ошибки и сохранением одного скрытого слоя приводит к снижению качества модели по сравнению с моделью, использующей только леммы. При этом добавление морфологических характеристик на вход сети, также не приводит к улучшению предсказания лемм. Данный результат можно объяснить необходимостью усложнения архитектуры сети (например, использования dropout-регуляризации) или более тщательным подбором гиперпарамтеров, что было затруднено ввиду временных затрат на обучение модели.

В результате была предложена модель отдельного классификатора морфологии с помощью сверточной нейронной сети. При сохранении архитектуры сети для каждой изменяемой части речи русского языка была натренирована отдельная модель.

Рекуррентная модель для предсказания лемм и сверточная модель для предсказания морфологии были использованы для ранжирования гипотез, возвращаемых системой распознавания Kaldi. Для этого были натренированы акустическая модель русского языка и 3-граммная модель со сглаживанием Кнесера-Нея.

Эксперимент показал, что рекуррентная модель на леммах, обученная на новостном корпусе в 2 млн. словоупотреблений не приводит к улучшению качества распознавания в комбинации с 3-граммной моделью со сглаживанием Кнесера-Нея, обученной на корпусе в 20 млн. словоупотреблений. Большой объем обучающего корпуса в последнем случае был необходим для обеспечения приемлемого качества первичных гипотез распознавания. Таким образом, необходима рекуррентная модель с большим размером словаря, обученная на корпусе большего размера и с более широким охватом тем, чем у новостного корпуса.

Использование сверточной сети для предсказания морфологии привело к снижению уровня пословной ошибки с 17.8% до 17.1%, причем есть основания считать, что модель может быть далее дообучена, а результаты улучшены без модификации модели и дополнительных обучающих данных.

Необходимо проведение дальнейших экспериментов с комбинированием рекуррентной модели на леммах, тематическим моделированием и моделями морфологии.

Для экспериментов с рекуррентной моделью была реализована программа на языке C++ с ипользованием библиотеки CUDA C для реализации вычислений на GPU.

В дальнейшем также планируется развитие рекуррентной языковой модели русского языка с использованием слоев LSTM и различных модификаций канонической модели, направленных на улучшение производительности, таких как иерархический softmax-слой.

Литература

1. *Кудинов Михаил Сергеевич.* Использование рекуррентных нейронных сетей для ранжирования списка гипотез в системах распознавания речи // *Современная наука: актуальные проблемы теории и практики. Серия «Естественные и технические науки.* — 2016. — Pp. 52–57.

2. *Kudinov Mikhail, Romanenko Alexander.* Hybrid language model based on recurrent neural network and probabilistic topic modeling // *Pattern Recognition and Image Analysis.* — 2016. — Vol. 26. — Pp. 587–592.

3. *Kudinov M.* Recurrent Neural Networks for Hypotheses Re-Scoring // Speech and Computer - 17th International Conference, SPECOM 2015, Athens, Greece, September 20-24, 2015, Proceedings. — 2015. — Pp. 341–347.

4. *Ray S.* Limits on the Application of Frequency-Based Language Models to OCR // ICDAR. — IEEE Computer Society, 2011. — Pp. 538–542.

5. Large language models in machine translation / T. Brants, A. Popat, P. Xu et al. // In EMNLP. — 2007. — Pp. 858–867.

6. *Jelinek F.* Statistical Methods for Speech Recognition. — Cambridge, MA, USA: MIT Press, 1997.

7. *Bahl L., Jelinek F., Mercer R.* Readings in Speech Recognition / Ed. by Alex Waibel, Kai-Fu Lee. — San Francisco, CA, USA, 1990. — Pp. 308–319.

8. *Bishop C.* Pattern Recognition and Machine Learning (Information Science and Statistics). — Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

9. *Huang X., Acero A., Hon H.-W.* Spoken Language Processing: A Guide to Theory, Algorithm, and System Development. — 1st edition. — Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001.

10. *Ney H.* Corpus-Based Statistical Methods in Speech and Language Processing / Ed. by G. Bloothooft, S. Young (eds.). — Dordrecht, The Netherlands: Kluwer Academic Publishers, 1997.

11. *Goodman J.* A Bit of Progress in Language Modeling // *Comput. Speech Lang.* — 2001. — Vol. 15, no. 4. — Pp. 403–434.

12. *Вернер М.* Основы кодирования. — М.: Техносфера, 2006.

13. *Mikolov T.* Statistical Language Models Based on Neural Networks: Ph.D. thesis. — 2012. — P. 129. http://www.fit.vutbr.cz/research/view_pub.php?id=10158.

14. *Левенштейн В.И.* Двоичные коды с исправлением выпадений, вставок и замещений символов // *Доклады Академий Наук СССР.* — 1965.

15. *Jelinek F., Mercer R.* Interpolated estimation of Markov source parameters from sparse data // Proceedings, Workshop on Pattern Recognition in Practice / Ed. by Edzard S. Gelsema, Laveen N. Kanal. — Amsterdam: North Holland, 1980. — Pp. 381–397.

16. *Katz S.* Estimation of probabilities from sparse data for the language model component of a speech recognizer // IEEE Transactions on Acoustics, Speech and Signal Processing. — 1987. — Pp. 400–401.

17. *Kneser R.and Ney H.* Improved backing-off for m-gram language modeling // In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing. — Vol. I. — Detroit, Michigan: 1995. — Pp. 181–184.

18. *Chen S., Goodman J.* An Empirical Study of Smoothing Techniques for Language Modeling // Proceedings of the 34th Annual Meeting on Association for Computational Linguistics. — ACL '96. — Stroudsburg, PA, USA: Association for Computational Linguistics, 1996. — Pp. 310–318.

19. *Whittaker E.* Statistical language modelling for automatic speech recognition of Russian and English: Ph.D. thesis.

20. *Rosenfeld R.* A maximum entropy approach to adaptive statistical language modelling // Computer Speech and Language. — Vol. 10. — 1996.

21. An Overview of the SPHINX-II Speech Recognition System / X. Huang, F. Alleva, M.-Y. Hwang, R. Rosenfeld // Proceedings of the Workshop on Human Language Technology. — HLT '93. — Stroudsburg, PA, USA: Association for Computational Linguistics, 1993. — Pp. 81–86.

22. *Ney H., Essen U., Kneser R.* On Structuring Probabilistic Dependencies in Stochastic Language Modelling // *Computer Speech and Language.* — 1994. — Vol. 8. — Pp. 1–38.

23. A Statistical Approach to Machine Translation / P. Brown, J. Cocke, S. A. Della Pietra et al. // *Comput. Linguist.* — 1990. — Vol. 16, no. 2. — Pp. 79–85.

24. *Bellegarda J.* Exploiting latent semantic information in statistical language modeling // *Proceedings of the IEEE*. — 2000. — Aug. — Vol. 88, no. 8. — Pp. 1279–1296.

25. Language model adaptation using latent dirichlet allocation and an efficient topic inference algorithm. / A. Heidel, H. Chang, L. Lee et al.

26. *Hofmann T.* Learning the Similarity of Documents: An Information-Geometric Approach to Document Retrieval and Categorization. — 2000.

27. Application of topic tracking model to language model adaptation and meeting analysis / S. Watanabe, T. Iwata, T. Hori et al. // Spoken Language Technology Workshop (SLT), 2010 IEEE. — 2010. — Dec. — Pp. 378–383.

28. *Mikolov T., Zweig G.* Context dependent recurrent neural network language model. // SLT. — IEEE, 2012. — Pp. 234–239.

29. A Dynamic Language Model for Speech Recognition / F. Jelinek, B. Merialdo, S. Roukos, M. Strauss // Proceedings of the Workshop on Speech and Natural Language. — HLT '91. — Stroudsburg, PA, USA: Association for Computational Linguistics, 1991. — Pp. 293–295.

30. *Kuhn R., Mori R. De.* A cache-based natural language model for speech recognition // *IEEE Transactions on Pattern Analysis and Machine Intelligence*. — 1990. — Jun. — Vol. 12, no. 6. — Pp. 570–583.

31. *Lau R., Rosenfeld R., Roukos S.* Trigger-based language models: a maximum entropy approach // Acoustics, Speech, and Signal Processing, 1993. ICASSP-93., 1993 IEEE International Conference on. — Vol. 2. — 1993. — April. — Pp. 45–48 vol.2.

32. *Iyer R. M., Ostendorf M.* Modeling long distance dependence in language: topic mixtures versus dynamic cache models // *IEEE Transactions on Speech and Audio Processing*. — 1999. — Jan. — Vol. 7, no. 1. — Pp. 30–39.

33. *Iyer R.* Language Modeling With Sentence-Level Mixtures. — 1994.

34. *Charniak E.* BLLIP 1987-89 WSJ Corpus Release 1. — 2000.

35. *Rosenfeld R.* Adaptive statistical language modeling: A maximum entropy approach: Ph.D. thesis / Department of the Navy, Naval Research Laboratory. — 2005.

36. *Jurafsky D., Martin J.* Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. — 1st edition. — Upper Saddle River, NJ, USA: Prentice Hall PTR, 2000.

37. *K. Seymore S. Chen R. Rosenfeld.* Nonlinear interpolation of topic models for language model adaptation // The 5th International Conference on Spoken Language Processing. — 1998.

38. *Coccaro N., Jurafsky D.* Towards Better Integration Of Semantic Predictors In Statistical Language Modeling // In Proceedings of ICSLP-98. — 1998. — Pp. 2403–2406.

39. *Воронцов К.В., Потапенко А.* Модификации EM-алгоритма для вероятностного тематического моделирования // *Машинное обучение и анализ данных.* — 2013. — Т. 1, № 6. — С. 657–686.

40. *Manning C., Schütze H.* Foundations of Statistical Natural Language Processing. — Cambridge, MA, USA: MIT Press, 1999.

41. *Тестелец Я.Г.* Введение в общий синтаксис. — М.: РГГУ, 2001. — 798 pp.

42. *Плунгян В.А.* Общая морфология. Введение в проблематику. — М.: УРСС, 2003. — 384 pp.

43. Self-organized language modeling for speech recognition / F. Jelinek, B. Merialdo, S. Roukos, M. Strauss I // Readings in Speech Recognition. — Morgan Kaufmann, 1990. — Pp. 450–506.

44. *Heeman P.* POS Tags and Decision Trees for Language Modeling // Proceedings of ACL'99. — 1999. — Pp. 129–137.

45. *Creutz M., Lagu M.* Unsupervised Discovery of Morphemes // Proceedings of the Workshop on Morphological and Phonological Learning of ACL-02. — 2002. — Pp. 21–30.

46. *V. Siivola T. Hirsimaki M. Creutz, Kurimo M.* Unlimited Vocabulary Speech Recognition Based on Morphs Discovered in an Unsupervised Manner // Proceedings of Eurospeech'03, Geneva (Switzerland). — 2003. — Pp. 2293–2296.

47. *Ratnaparkhi A.* A Maximum Entropy Model for Part-of-Speech Tagging // Proceedings of the Conference on Empirical Methods in Natural Language Processing / Ed. by Eric Brill, Kenneth Church. — University of Pennsylvania, Philadelphia, PA: Association for Computational Linguistics, 1996. — Pp. 133–142.

48. *Lafferty J.* Conditional random fields: Probabilistic models for segmenting and labeling sequence data. — Morgan Kaufmann, 2001. — Pp. 282–289.

49. *Nivre J.* Algorithms for Deterministic Incremental Dependency Parsing. // *Computational Linguistics.* — 2008. — Vol. 34, no. 4. — Pp. 513–553.

50. *Музычка С., Пионтковская И.* Графовый подход в задаче построения синтаксических деревьев для русского языка // Компьютерная лингвистика и интеллектуальные технологии. — Vol. 1. — РГГУ, 2015. — Pp. 468–474.

51. *F. Jelinek J.D. Lafferty, Mercer R.L.* Basic Methods of Probabilistic Context Free Grammars // Speech Recognition and Understanding: Recent Advances, Trends, and Applications. — Vol. 75. — Computer and Systems Sciences, 1992. — Pp. 345–360.

52. *M. Popel D. Mareček.* Perplexity of n-gram and dependency language models // Text, Speech and Dialogue. — Springer Berlin Heidelberg, 2010. — Pp. 173–180.

53. *Kirchhoff K. et al.* Novel Speech Recognition Models for Arabic. — 2002.

54. *Bilmes J., Kirchhoff K.* Factored Language Models and Generalized Parallel Backoff // Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology. — Vol. 2 of *NAACL-Short '03*. — Stroudsburg, PA, USA: Association for Computational Linguistics, 2003. — Pp. 4–6.

55. *Koller D., Friedman N.* Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning. — The MIT Press, 2009.

56. *Oparin I.* Language models for automatic speech recognition of inflectional languages: Ph.D. thesis / University of West Bohemia. — 2008.

57. Recurrent neural network based language model / T. Mikolov, M. Karafiát, L. Burget et al. // INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010. — 2010. — Pp. 1045–1048.

58. *Bengio Y., Ducharme R., Vincent P.* A Neural Probabilistic Language Model // Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems (NIPS) 2000, Denver, CO, USA. — 2000. — Pp. 932–938.

59. *Elman J.* Finding structure in time // *COGNITIVE SCIENCE*. — 1990. — Vol. 14, no. 2. — Pp. 179–211.

60. *Cruse H.* Neural Networks as Cybernetic Systems - Part II // *Brains, Minds, and Media*. — 2006. — no. 1.

61. *Hopfield J.* Neural networks and physical systems with emergent collective computational abilities // *Proceedings of the National Academy of Sciences of the United States of America*. — 1982. — Vol. 79, no. 8. — Pp. 2554–2558.

62. How to Construct Deep Recurrent Neural Networks. / R. Pascanu, Ç. Gülçehre, K. Cho, Y. Bengio // *CoRR*. — 2013. — Vol. abs/1312.6026.

63. *Timmaraju A., Khanna V.* Sentiment Analysis on Movie Reviews using Recursive and Recurrent Neural Network Architectures.

64. *Pascanu R.* On Recurrent and Deep Neural Networks: Ph.D. thesis / Université de Montreal.

65. *Williams R., Peng J.* An Efficient Gradient-Based Algorithm for On-Line Training of Recurrent Network Trajectories // *Neural Computation*. — 1990. — Vol. 2. — Pp. 490–501.

66. *Bengio Y., Simard P., Frasconi P.* Learning Long-term Dependencies with Gradient Descent is Difficult // *Trans. Neur. Netw.* — 1994. — Vol. 5, no. 2. — Pp. 157–166.

67. *Hochreiter S., Schmidhuber J.* Long Short-Term Memory // *Neural Comput.* — 1997. — Vol. 9, no. 8. — Pp. 1735–1780.

68. *Pascanu R., Mikolov T., Bengio Y.* On the difficulty of training recurrent neural networks // Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013. — 2013. — Pp. 1310–1318.

69. *Martens J., Sutskever I.* Training Deep and Recurrent Networks with Hessian-Free Optimization // Neural Networks: Tricks of the Trade - Second Edition. — 2012. — Pp. 479–535.

70. *Karpathy A., Li Fei-Fei.* Deep visual-semantic alignments for generating image descriptions // IEEE Conference on Computer Vision and Pattern Recognition 2015, Boston, MA, USA, June 7-12, 2015. — 2015. — Pp. 3128–3137.

71. *Zaremba W., Sutskever I., Vinyals O.* Recurrent Neural Network Regularization // *CoRR.* — 2014. — Vol. abs/1409.2329.

72. *Vazhenina D., Markov K.* Evaluation of Advanced Language Modeling Techniques for Russian LVCSR. — 2013. — Pp. 124–131.

73. RNNLM - Recurrent Neural Network Language Modeling Toolkit / T. Mikolov, S. Kombrink, A. Deoras et al. — 2011.

74. *S. Muzychka, A. Romanenko, I. Piontkovskaya.* Conditional Random Field for morphological disambiguation in Russian // Computational Linguistics and Intellectual Technologies. — No. 13. — РГГУ, 2014. — Pp. 456–466.

75. *J. Thorsten.* Optimizing Search Engines Using Clickthrough Data // Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. — 2002. — Pp. 133–142.

76. Morphology-Based Language Modeling for Arabic Speech Recognition / D. Vergyri, K. Kirchhoff, K. Duh, A. Stolcke // In Proc. of ICSLP. — 2004. — Pp. 2245–2248.

77. *Serge Sharoff Mikhail Kopotev Tomaz Erjavec Anna Feldman, Divjak Dagmar.* Designing and Evaluating a Russian Tagset // Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08). — Marrakech, Morocco: European Language Resources Association (ELRA), 2008. — may.

78. Multilingual Speech Data Collection for the Assessment of Pronunciation and Prosody in a Language Learning System / O. Jokisch, A. Wagner, R. Sabo et al. — 2009. — Pp. 515–520.

79. Indexing by latent semantic analysis / S. Deerwester, S. Dumais, G. Furnas et al. // *Journal of the American Society for Information Science*. — 1990. — Vol. 41, no. 6. — Pp. 391–407.

80. *Mikolov T., Sutskever I. et al.* Distributed Representations of Words and Phrases and their Compositionality // Advances in Neural Information Processing Systems. — 2013. — Pp. 3111–3119.

81. *Mikolov T., Le Quoc V., Sutskever I.* Exploiting Similarities among Languages for Machine Translation // *CoRR*. — 2013. — Vol. abs/1309.4168.

82. *Mikolov T., Chen K. et al.* Efficient Estimation of Word Representations in Vector Space // *CoRR*. — 2013. — Vol. abs/1301.3781.

83. Solving Verbal Comprehension Questions in IQ Test by Knowledge-Powered Word Embedding / H. Wang, B. Gao, J. Bian et al. // *CoRR*. — 2015. — Vol. abs/1505.07909.

84. *Goldberg Y., Levy O.* word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method // *CoRR*. — 2014. — Vol. abs/1402.3722.

85. *Levy O., Goldberg Y.* Neural Word Embedding as Implicit Matrix Factorization // Advances in Neural Information Processing Systems 27 / Ed. by Z. Ghahramani, M. Welling, C. Cortes et al. — 2014. — Pp. 2177–2185.

86. *Kneser R., Steinbiss V.* On the dynamic adaptation of stochastic language models // Acoustics, Speech, and Signal Processing, 1993. ICASSP-93., 1993 IEEE International Conference on / IEEE. — Vol. 2. — 1993. — Pp. 586–589.

87. *Khudanpur S., Wu J.* Maximum Entropy Techniques for Exploiting Syntactic, Semantic and Collocational Dependencies in Language Modeling.

88. *Chen S.* Performance Prediction for Exponential Language Models // Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics. — NAACL '09. — Stroudsburg, PA, USA: Association for Computational Linguistics, 2009. — Pp. 450–458.

89. *Hofmann T.* Probabilistic Latent Semantic Indexing. — 1999. — Pp. 50–57.

90. Statistical topic models for multilabel document classification / T. Rubin, A. Chambers, P. Smyth, M. Steyvers // *Machine Learning*. — 2012. — Vol. 88, no. 1-2. — Pp. 157–208.

91. *Yi X., Allan J.* A comparative study of utilizing topic models for information retrieval // *Advances in Information Retrieval*. — 2009.

92. *Yeh J.-h., Wu M.-l.* Recommendation based on latent topics and social network analysis // Proceedings of the 2010 Second International Conference on Computer Engineering and Applications. — Vol. 1. — IEEE Computer Society, 2010. — Pp. 209–213.

93. *Воронцов К.В.* Вероятностное тематическое моделирование. — 2013.

94. *Dempster A., Laird N., Rubin D.* Maximum likelihood from incomplete data via the EM algorithm // *Journal of the Royal Statistical Society, Series B.* — 1977. — Vol. 39, no. 1. — Pp. 1–38.

95. *Воронцов К.В., Потапенко А.* Additive Regularization of Topic Models. http://machinelearning.ru/wiki/images/4/47/Voron14mlj.pdf.

96. *Blei David M., Ng Andrew Y., Jordan Michael I.* Latent Dirichlet Allocation // *J. Mach. Learn. Res.* — 2003. — Vol. 3. — Pp. 993–1022.

97. *Wallach Hanna M., Mimno David M., McCallum Andrew.* Rethinking LDA: Why Priors Matter // NIPS. — Curran Associates, Inc., 2009. — Pp. 1973–1981.

98. *Воронцов К.В., Потапенко А.* Регуляризация, робастность и разреженность вероятностных тематических моделей, Компьютерные исследования и моделирование // Компьютерная лингвитика и интеллектуальные технологии. — № 13. — РГГУ, 2014.

99. *Goldberg Yoav.* A Primer on Neural Network Models for Natural Language Processing // *CoRR.* — 2015. — Vol. abs/1510.00726. http://arxiv.org/abs/1510.00726.

100. *Simonyan Karen, Zisserman Andrew.* Very Deep Convolutional Networks for Large-Scale Image Recognition // *CoRR.* — 2014. — Vol. abs/1409.1556. http://arxiv.org/abs/1409.1556.

101. *Szegedy Christian, Liu Wei et al.* Going Deeper with Convolutions // *CoRR.* — 2014. — Vol. abs/1409.4842.

102. *Karpathy Andrej et al.* Large-Scale Video Classification with Convolutional Neural Networks // Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition. — CVPR '14. — 2014. — Pp. 1725–1732.

103. *Abdel-Hamid Ossama, Mohamed Abdel-Rahman ohers.* Convolutional Neural Networks for Speech Recognition // *IEEE/ACM Trans. Audio, Speech and Lang. Proc.* — 2014. — Vol. 22, no. 10. — Pp. 1533–1545.

104. Gradient-based learning applied to document recognition / Y. Lecun, L. Bottou, Y. Bengio, P. Haffner // *Proceedings of the IEEE.* — 1998. — Vol. 86, no. 11. — Pp. 2278–2324.

105. *Kim Yoon.* Convolutional Neural Networks for Sentence Classification // *CoRR.* — 2014. — Vol. abs/1408.5882. http://arxiv.org/abs/1408.5882.

106. *Kalchbrenner Nal, Grefenstette Edward, Blunsom Phil.* A Convolutional Neural Network for Modelling Sentences // *CoRR.* — 2014. — Vol. abs/1404.2188. http://arxiv.org/abs/1404.2188.

107. *Арефьев Н.В., Панченко А.И. и др.* Сравнение трех систем семантической близости для русского языка // Компьютерная лингвитика и интеллектуальные технологии. — № 14. — РГГУ, 2015.

108. Caffe: Convolutional Architecture for Fast Feature Embedding / Yangqing Jia, Evan Shelhamer, Donahue et al. // *arXiv preprint arXiv:1408.5093*. — 2014.

109. The Kaldi Speech Recognition Toolkit / Daniel Povey, Arnab Ghoshal, Boulianne et al. // IEEE 2011 Workshop on Automatic Speech Recognition and Understanding. — IEEE Signal Processing Society, 2011.

110. *Mohri Mehryar*. Finite-State Transducers in Language and Speech Processing // *Computational Linguistics*. — 1997. — Vol. 23. — Pp. 269–311.

111. *Mohri Mehryar, Pereira Fernando, Riley Michael*. Weighted Finite-State Transducers in Speech Recognition. — 2001.

112. База речевых фрагментов русского языка «ISABASE / Д. С. Богданов, О. Ф. Кривнова, А. Я. Подрабинович, В. В. Фарсобина // Интеллектуальные технологии ввода и обработки информации. — Эдиторил УРСС Москва, 1998. — С. 74–85.

113. *Stolcke Andreas*. SRILM - An Extensible Language Modeling Toolkit. — 2002. — Pp. 901–904.

114. *Bisani Maximilian, Ney Hermann*. Joint-sequence Models for Grapheme-to-phoneme Conversion // *Speech Commun.* — 2008. — Vol. 50, no. 5. — Pp. 434–451.

Список рисунков

Список таблиц