

Deep contextualized word representations	Глубокие контекстуальные представления слов
Matthew E. Peters† , Mark Neumann† , Mohit Iyyer† , Matt Gardner†, {matthewp,markn,mohiti,mattg}@allenai.org	
Christopher Clark * , Kenton Lee * , Luke Zettlemoyer† * {csquared,kentonl,lsz}@cs.washington.edu	
Allen Institute for Artificial Intelligence * Paul G. Allen School of Computer Science & Engineering, University of Washington	
Abstract	Аннотация
We introduce a new type of deep contextualized word representation that models both (1) complex characteristics of word use (e.g., syntax and semantics), and (2) how these uses vary across linguistic contexts (i.e., to model polysemy). Our word vectors are learned functions of the internal states of a deep bidirectional language model (biLM), which is pretrained on a large text corpus. We show that these representations can be easily added to existing models and significantly improve the state of the art across six challenging NLP problems, including question answering, textual entailment and sentiment analysis. We also present an analysis showing that exposing the deep internals of the pre-trained network is crucial, allowing downstream models to mix different types of semi-supervision signals.	Мы вводим новый тип глубокого контекстуального представления слов, который моделирует как (1) сложные характеристики употребления слов (например, синтаксис и семантику), так и (2) то, как эти употребления варьируются в зависимости от языкового контекста (т. е. для моделирования полисемии). Наши векторы слов являются выученными функциями внутренних состояний модели глубокого двунаправленного языка (biLM), которая предварительно обучена на большом текстовом корпусе. Мы показываем, что эти представления можно легко добавить к существующим моделям и значительно улучшить состояние дел в шести сложных задачах НЛП, включая ответы на вопросы, текстовые следствия и анализ настроений. Мы также представляем анализ, показывающий, что раскрытие глубоких внутренностей предварительно обученной сети имеет решающее значение, позволяя нижестоящим моделям смешивать различные типы сигналов полуконтроля.
1 Introduction	1 Введение
Pre-trained word representations (Mikolov et al., 2013; Pennington et al., 2014) are a key component in many neural language understanding models. However, learning high quality representations can be challenging. They should ideally model both (1) complex characteristics of word use (e.g., syntax and semantics), and (2) how these uses vary across linguistic contexts (i.e., to model polysemy). In this paper, we introduce a new type of deep contextualized word representation that directly addresses both challenges, can be easily integrated into existing models, and significantly improves the state of the art in every considered case across a range of challenging language understanding problems.	Предварительно обученные представления слов (Миколов и др., 2013; Пеннингтон и др., 2014) являются ключевым компонентом многих моделей нейронного понимания языка. Однако изучение высококачественных представлений может быть сложной задачей. В идеале они должны моделировать как (1) сложные характеристики употребления слов (например, синтаксис и семантику), так и (2) то, как эти употребления меняются в зависимости от языкового контекста (т. е. моделировать полисемию). В этой статье мы представляем новый тип глубокого контекстуального представления слов, который напрямую решает обе проблемы, может быть легко интегрирован в существующие модели и значительно улучшает состояние дел в каждом рассматриваемом случае в ряде сложных проблем понимания языка.
Our representations differ from traditional word type embeddings in that each token is assigned a representation that is a function of the entire input	Наши представления отличаются от традиционных вложений типов слов тем, что каждому токenu назначается представление, которое является

<p>sentence. We use vectors derived from a bidirectional LSTM that is trained with a coupled language model (LM) objective on a large text corpus. For this reason, we call them ELMo (Embeddings from Language Models) representations. Unlike previous approaches for learning contextualized word vectors (Peters et al., 2017; McCann et al., 2017), ELMo representations are deep, in the sense that they are a function of all of the internal layers of the biLM. More specifically, we learn a linear combination of the vectors stacked above each input word for each end task, which markedly improves performance over just using the top LSTM layer.</p>	<p>функцией всего входного предложения. Мы используем векторы, полученные из двунаправленного LSTM, который обучается с помощью модели связанного языка (LM) на большом текстовом корпусе. По этой причине мы называем их представлениями ELMo (Embeddings from Language Models). В отличие от предыдущих подходов к изучению контекстуализированных векторов слов (Peters et al., 2017; McCann et al., 2017), представления ELMo являются глубокими в том смысле, что они являются функцией всех внутренних слоев biLM. В частности, мы изучаем линейную комбинацию векторов, сложенных над каждым входным словом для каждой конечной задачи, что заметно повышает производительность по сравнению с использованием только верхнего слоя LSTM.</p>
<p>Combining the internal states in this manner allows for very rich word representations. Using intrinsic evaluations, we show that the higher-level LSTM states capture context-dependent aspects of word meaning (e.g., they can be used without modification to perform well on supervised word sense disambiguation tasks) while lowerlevel states model aspects of syntax (e.g., they can be used to do part-of-speech tagging). Simultaneously exposing all of these signals is highly beneficial, allowing the learned models select the types of semi-supervision that are most useful for each end task.</p>	<p>Объединение внутренних состояний таким образом позволяет очень богато представлять слова. Используя внутренние оценки, мы показываем, что состояния LSTM более высокого уровня охватывают контекстно-зависимые аспекты значения слова (например, их можно использовать без модификации для эффективного выполнения контролируемых задач устранения неоднозначности смысла слова), в то время как состояния более низкого уровня моделируют аспекты синтаксиса (например, их можно использовать для маркировки частей речи). Одновременная демонстрация всех этих сигналов очень полезна, поскольку позволяет обученным моделям выбирать типы полуконтроля, которые наиболее полезны для каждой конечной задачи.</p>
<p>Extensive experiments demonstrate that ELMo representations work extremely well in practice. We first show that they can be easily added to existing models for six diverse and challenging language understanding problems, including textual entailment, question answering and sentiment analysis. The addition of ELMo representations alone significantly improves the state of the art in every case, including up to 20% relative error reductions. For tasks where direct comparisons are possible, ELMo outperforms CoVe (McCann et al., 2017), which computes contextualized representations using a neural machine translation encoder. Finally, an analysis of both ELMo and CoVe reveals that deep representations outperform arXiv:1802.05365v2 [cs.CL] 22 Mar 2018 those derived from just the top layer of an LSTM. Our trained models and code are publicly available, and we expect that ELMo will provide similar gains for many other NLP problems.¹</p>	<p>Обширные эксперименты показывают, что представления ELMo очень хорошо работают на практике. Сначала мы показываем, что их можно легко добавить к существующим моделям для решения шести разнообразных и сложных проблем понимания языка, включая вывод текста, ответы на вопросы и анализ настроений. Добавление представлений ELMo само по себе значительно улучшает уровень техники в каждом случае, включая снижение относительной ошибки до 20%. В задачах, где возможно прямое сравнение, ELMo превосходит CoVe (McCann et al., 2017), который вычисляет контекстуализированные представления с помощью кодировщика нейронного машинного перевода. Наконец, анализ как ELMo, так и CoVe показывает, что глубокие представления превосходят arXiv:1802.05365v2 [cs.CL] 22 марта 2018 г. те, которые получены только из верхнего уровня LSTM. Наши обученные модели и код общедоступны, и мы ожидаем, что ELMo</p>

	обеспечит аналогичные преимущества для многих других задач НЛП.1
2 Related work	2 Связанные работы
Due to their ability to capture syntactic and semantic information of words from large scale unlabeled text, pretrained word vectors (Turian et al., 2010; Mikolov et al., 2013; Pennington et al., 2014) are a standard component of most state-of-the-art NLP architectures, including for question answering (Liu et al., 2017), textual entailment (Chen et al., 2017) and semantic role labeling (He et al., 2017). However, these approaches for learning word vectors only allow a single contextindependent representation for each word.	Предварительно обученные векторы слов (Turian et al., 2010; Mikolov et al., 2013; Pennington et al., 2014) извлекают синтаксическую и семантическую информацию о словах из крупномасштабного неразмеченного текста. - современные архитектуры НЛП, в том числе для ответов на вопросы (Liu et al., 2017), текстового следования (Chen et al., 2017) и обозначения семантических ролей (He et al., 2017). Однако эти подходы к изучению векторов слов допускают только одно контекстно-независимое представление для каждого слова.
Previously proposed methods overcome some of the shortcomings of traditional word vectors by either enriching them with subword information (e.g., Wieting et al., 2016; Bojanowski et al., 2017) or learning separate vectors for each word sense (e.g., Neelakantan et al., 2014). Our approach also benefits from subword units through the use of character convolutions, and we seamlessly incorporate multi-sense information into downstream tasks without explicitly training to predict predefined sense classes.	Предложенные ранее методы преодолевают некоторые недостатки традиционных векторов слов, либо обогащая их информацией о подсловах (например, Wieting et al., 2016; Bojanowski et al., 2017), либо изучая отдельные векторы для каждого значения слова (например, Neelakantan et al. ., 2014). Наш подход также выигрывает от единиц подслов за счет использования сверток символов, и мы плавно включаем многосмысловую информацию в последующие задачи без явного обучения прогнозированию предопределенных классов смыслов.
Other recent work has also focused on learning context-dependent representations. context2vec (Melamud et al., 2016) uses a bidirectional Long Short Term Memory (LSTM; Hochreiter and Schmidhuber, 1997) to encode the context around a pivot word. Other approaches for learning contextual embeddings include the pivot word itself in the representation and are computed with the encoder of either a supervised neural machine translation (MT) system (CoVe; McCann et al., 2017) or an unsupervised language model (Peters et al., 2017). Both of these approaches benefit from large datasets, although the MT approach is limited by the size of parallel corpora. In this paper, we take full advantage of access to plentiful monolingual data, and train our biLM on a corpus with approximately 30 million sentences (Chelba et al., 2014). We also generalize these approaches to deep contextual representations, which we show work well across a broad range of diverse NLP tasks.	Другая недавняя работа также была сосредоточена на изучении контекстно-зависимых представлений. context2vec (Melamud et al., 2016) использует двунаправленную долговременную кратковременную память (LSTM; Hochreiter and Schmidhuber, 1997) для кодирования контекста вокруг опорного слова. Другие подходы к обучению контекстуальным встраиваниям включают в себя само опорное слово в представлении и вычисляются с помощью кодировщика либо системы контролируемого нейронного машинного перевода (MT) (CoVe; McCann et al., 2017), либо неконтролируемой языковой модели (Peters et al. ., 2017). Оба этих подхода выигрывают от больших наборов данных, хотя подход машинного перевода ограничен размером параллельных корпусов. В этой статье мы в полной мере используем доступ к обильным одноязычным данным и обучаем наш biLM на корпусе, содержащем примерно 30 миллионов предложений (Челба и др., 2014). Мы также обобщаем эти подходы к глубоким контекстуальным представлениям, которые, как мы показываем, хорошо работают в широком диапазоне разнообразных задач НЛП.
Previous work has also shown that different layers of deep biRNNs encode different types of information. For example, introducing multi-task syntactic	Предыдущая работа также показала, что разные уровни глубоких biRNN кодируют разные типы информации. Например, введение

<p>supervision (e.g., part-of-speech tags) at the lower levels of a deep LSTM can improve overall performance of higher level tasks such as dependency parsing (Hashimoto et al., 2017) or CCG super tagging (Søgaard and Goldberg, 2016). In an RNN-based encoder-decoder machine translation system, Belinkov et al. (2017) showed that the representations learned at the first layer in a 2- layer LSTM encoder are better at predicting POS tags than second layer. Finally, the top layer of an LSTM for encoding word context (Melamud et al., 2016) has been shown to learn representations of word sense. We show that similar signals are also induced by the modified language model objective of our ELMo representations, and it can be very beneficial to learn models for downstream tasks that mix these different types of semi-supervision.</p>	<p>многозадачного синтаксического контроля (например, тегов частей речи) на нижних уровнях глубокого LSTM может повысить общую производительность задач более высокого уровня, таких как анализ зависимостей (Hashimoto et al., 2017) или CCG super. мечение (Søgaard and Goldberg, 2016). В системе машинного перевода кодер-декодер на основе RNN Белинков и др. (2017) показали, что представления, полученные на первом уровне в двухуровневом кодере LSTM, лучше предсказывают POS-теги, чем на втором уровне. Наконец, было показано, что верхний уровень LSTM для кодирования контекста слова (Melamud et al., 2016) изучает представления смысла слова. Мы показываем, что подобные сигналы также индуцируются измененной целью языковой модели наших представлений ELMo, и может быть очень полезно изучить модели для последующих задач, которые смешивают эти различные типы полуконтроля.</p>
<p>Dai and Le (2015) and Ramachandran et al. (2017) pretrain encoder-decoder pairs using language models and sequence autoencoders and then fine tune with task specific supervision. In contrast, after pretraining the biLM with unlabeled data, we fix the weights and add additional task-specific model capacity, allowing us to leverage large, rich and universal biLM representations for cases where downstream training data size dictates a smaller supervised model.</p>	<p>Дай и Ле (2015) и Ramachandran et al. (2017) предварительно обучают пары кодировщик-декодер с использованием языковых моделей и автокодировщиков последовательности, а затем выполняют точную настройку с контролем конкретной задачи. Напротив, после предварительного обучения biLM с неразмеченными данными мы фиксируем веса и добавляем дополнительную емкость модели для конкретных задач, что позволяет нам использовать большие, богатые и универсальные представления biLM для случаев, когда размер данных для последующего обучения диктует меньшую контролируемую модель.</p>
<p>3 ELMo: Embeddings from Language Models</p>	<p>3 ELMo: встраивания из языковых моделей</p>
<p>Unlike most widely used word embeddings (Pennington et al., 2014), ELMo word representations are functions of the entire input sentence, as described in this section. They are computed on top of two-layer biLMs with character convolutions (Sec. 3.1), as a linear function of the internal network states (Sec. 3.2). This setup allows us to do semi-supervised learning, where the biLM is pretrained at a large scale (Sec. 3.4) and easily incorporated into a wide range of existing neural NLP architectures (Sec. 3.3).</p>	<p>В отличие от наиболее широко используемых вложений слов (Pennington et al., 2014), представления слов ELMo являются функциями всего входного предложения, как описано в этом разделе. Они вычисляются поверх двухслойных biLM со сверткой символов (раздел 3.1) как линейная функция состояний внутренней сети (раздел 3.2). Эта установка позволяет нам проводить полуконтролируемое обучение, при котором biLM предварительно обучается в больших масштабах (раздел 3.4) и легко интегрируется в широкий спектр существующих нейронных архитектур НЛП (раздел 3.3).</p>
<p>3.1 Bidirectional language models</p>	<p>3.1 Двухнаправленные языковые модели</p>
<p>Given a sequence of N tokens, (t_1, t_2, \dots, t_N), a forward language model computes the probability of the sequence by modeling the probability of token t_k given the history (t_1, \dots, t_{k-1}):</p>	<p>Для последовательности N маркеров (t_1, t_2, \dots, t_N) прямая языковая модель вычисляет вероятность последовательности путем моделирования вероятности маркера t_k с учетом истории (t_1, \dots, t_{k-1}):</p>

$p(t_1, t_2, \dots, t_N) = \prod_{k=1}^N p(t_k t_1, t_2, \dots, t_{k-1}).$	
<p>Recent state-of-the-art neural language models (J'ozefowicz et al., 2016; Melis et al., 2017; Merity et al., 2017) compute a context-independent token representation $x_{LM\ k}$ (via token embeddings or a CNN over characters) then pass it through L layers of forward LSTMs. At each position k, each LSTM layer outputs a context-dependent representation $\rightarrow h_{LM\ k,j}$ where $j = 1, \dots, L$. The top layer LSTM output, $\rightarrow h_{LM\ k,L}$, is used to predict the next token t_{k+1} with a Softmax layer.</p>	<p>Недавние современные нейронные языковые модели (J'ozefowicz et al., 2016; Melis et al., 2017; Merity et al., 2017) вычисляют контекстно-независимое представление токена $x_{LM\ k}$ (посредством встраивания токенов или CNN над символами), затем пропускают его через L уровней прямых LSTM. В каждой позиции k каждый уровень LSTM выводит контекстно-зависимое представление $\rightarrow h_{LM\ k,j}$, где $j = 1, \dots, L$. Выход LSTM верхнего уровня, $\rightarrow h_{LM\ k,L}$, используется для прогнозирования следующего токена t_{k+1} с помощью слоя Softmax.</p>
<p>A backward LM is similar to a forward LM, except it runs over the sequence in reverse, predicting the previous token given the future context:</p>	<p>Обратный LM похож на прямой LM, за исключением того, что он проходит последовательность в обратном порядке, предсказывая предыдущий токен с учетом будущего контекста:</p>
$p(t_1, t_2, \dots, t_N) = \prod_{k=1}^N p(t_k t_{k+1}, t_{k+2}, \dots, t_N).$	
<p>It can be implemented in an analogous way to a forward LM, with each backward LSTM layer j in a L layer deep model producing representations $\leftarrow h_{LM\ k,j}$ of t_k given (t_{k+1}, \dots, t_N). A biLM combines both a forward and backward LM. Our formulation jointly maximizes the log likelihood of the forward and backward directions:</p>	<p>Это может быть реализовано аналогично прямому LM, с каждым обратным уровнем LSTM j в глубокой модели L уровня, создающим представления $\leftarrow h_{LM\ k,j}$ заданного t_k (t_{k+1}, \dots, t_N). BiLM сочетает в себе как прямой, так и обратный LM. Наша формулировка совместно максимизирует логарифмическую вероятность прямого и обратного направлений:</p>
$\sum_{k=1}^N (\log p(t_k t_1, \dots, t_{k-1}; \Theta_x, \vec{\Theta}_{LSTM}, \Theta_s) + \log p(t_k t_{k+1}, \dots, t_N; \Theta_x, \overleftarrow{\Theta}_{LSTM}, \Theta_s))$	
<p>We tie the parameters for both the token representation (Θ_x) and Softmax layer (Θ_s) in the forward and backward direction while maintaining separate parameters for the LSTMs in each direction. Overall, this formulation is</p>	<p>Мы связываем параметры как для представления токена (Θ_x), так и для слоя Softmax (Θ_s) в прямом и обратном направлениях, сохраняя при этом отдельные параметры для LSTM в каждом направлении. В целом, эта</p>

similar to the approach of Peters et al. (2017), with the exception that we share some weights between directions instead of using completely independent parameters. In the next section, we depart from previous work by introducing a new approach for learning word representations that are a linear combination of the biLM layers.	формулировка аналогична подходу Peters et al. (2017), за исключением того, что мы делим некоторые веса между направлениями вместо того, чтобы использовать полностью независимые параметры. В следующем разделе мы отходим от предыдущей работы, представляя новый подход к изучению представлений слов, которые представляют собой линейную комбинацию слоев biLM.
3.2 ELMo	3.2 ELMo
ELMo is a task specific combination of the intermediate layer representations in the biLM. For each token t_k , a L-layer biLM computes a set of $2L + 1$ representations	ELMo представляет собой специфичную для задачи комбинацию представлений промежуточного уровня в biLM. Для каждого токена t_k biLM L-уровня вычисляет набор из $2L + 1$ представлений.
$R_k = \{\mathbf{x}_k^{LM}, \overrightarrow{\mathbf{h}}_{k,j}^{LM}, \overleftarrow{\mathbf{h}}_{k,j}^{LM} \mid j = 1, \dots, L\}$ $= \{\mathbf{h}_{k,j}^{LM} \mid j = 0, \dots, L\},$	
where $\mathbf{h}_{k,0}^{LM}$ is the token layer and $\mathbf{h}_{k,j}^{LM} = [\overrightarrow{\mathbf{h}}_{k,j}^{LM}; \overleftarrow{\mathbf{h}}_{k,j}^{LM}]$, for each biLSTM layer. For inclusion in a downstream model, ELMo collapses all layers in R into a single vector, $\text{ELMo}_k = E(R_k; \Theta_e)$. In the simplest case, ELMo just selects the top layer, $E(R_k) = \mathbf{h}_{k,L}^{LM}$, as in TagLM (Peters et al., 2017) and CoVe (McCann et al., 2017). More generally, we compute a task specific weighting of all biLM layers:	где $\mathbf{h}_{k,0}^{LM}$ — слой маркеров, а $\mathbf{h}_{k,j}^{LM} = [\overrightarrow{\mathbf{h}}_{k,j}^{LM}; \overleftarrow{\mathbf{h}}_{k,j}^{LM}]$, для каждого слоя biLSTM. Для включения в нисходящую модель ELMo сворачивает все слои в R в один вектор, $\text{ELMo}_k = E(R_k; \Theta_e)$. В простейшем случае ELMo просто выбирает верхний слой, $E(R_k) = \mathbf{h}_{k,L}^{LM}$, как в TagLM (Peters et al., 2017) и CoVe (McCann et al., 2017). В более общем смысле мы вычисляем взвешивание всех слоев biLM для конкретной задачи:
$\text{ELMo}_k^{task} = E(R_k; \Theta^{task}) = \gamma^{task} \sum_{j=0}^L s_j^{task} \mathbf{h}_{k,j}^{LM}. \quad (1)$	
In (1), s task are softmax-normalized weights and the scalar parameter γ^{task} allows the task model to scale the entire ELMo vector. γ is of practical importance to aid the optimization process (see supplemental material for details). Considering that the activations of each biLM layer have a different distribution, in some cases it also helped to apply layer normalization (Ba et al., 2016) to each biLM layer before weighting.	В (1) задача s представляет собой веса, нормализованные по softmax, а скалярный параметр γ^{task} позволяет модели задачи масштабировать весь вектор ELMo. γ имеет практическое значение для облегчения процесса оптимизации (подробности см. В дополнительных материалах). Учитывая, что активации каждого слоя biLM имеют различное распределение, в некоторых случаях это также помогло применить нормализацию слоя (Ba et al., 2016) к каждому слою biLM перед взвешиванием.
3.3 Using biLMs for supervised NLP tasks	3.3 Использование biLM для контролируемых задач НЛП

Given a pre-trained biLM and a supervised architecture for a target NLP task, it is a simple process to use the biLM to improve the task model. We simply run the biLM and record all of the layer representations for each word. Then, we let the end task model learn a linear combination of these representations, as described below.	Учитывая предварительно обученный biLM и контролируемую архитектуру для целевой задачи NLP, использование biLM для улучшения модели задачи является простым процессом. Мы просто запускаем biLM и записываем все представления слоев для каждого слова. Затем мы позволяем модели конечной задачи изучить линейную комбинацию этих представлений, как описано ниже.
First consider the lowest layers of the supervised model without the biLM. Most supervised NLP models share a common architecture at the lowest layers, allowing us to add ELMo in a consistent, unified manner. Given a sequence of tokens (t_1, \dots, t_N), it is standard to form a context-independent token representation x_k for each token position using pre-trained word embeddings and optionally character-based representations. Then, the model forms a context-sensitive representation h_k , typically using either bidirectional RNNs, CNNs, or feed forward networks.	Сначала рассмотрим самые нижние слои контролируемой модели без biLM. Большинство моделей НЛП с учителем имеют общую архитектуру на самых нижних уровнях, что позволяет нам добавлять ELMo согласованным и унифицированным образом. Для заданной последовательности токенов (t_1, \dots, t_N) стандартно формировать независимое от контекста представление x_k токенов для каждой позиции токенов с использованием предварительно обученных вложений слов и, необязательно, представлений на основе символов. Затем модель формирует контекстно-зависимое представление h_k , обычно используя двунаправленные RNN, CNN или сети прямой связи.
To add ELMo to the supervised model, we first freeze the weights of the biLM and then concatenate the ELMo vector $ELMo_{task\ k}$ with x_k and pass the ELMo enhanced representation $[x_k; ELMo_{task\ k}]$ into the task RNN. For some tasks (e.g., SNLI, SQuAD), we observe further improvements by also including ELMo at the output of the task RNN by introducing another set of output specific linear weights and replacing h_k with $[h_k; ELMo_{task\ k}]$. As the remainder of the supervised model remains unchanged, these additions can happen within the context of more complex neural models. For example, see the SNLI experiments in Sec. 4 where a bi-attention layer follows the biLSTMs, or the coreference resolution experiments where a clustering model is layered on top of the biLSTMs.	Чтобы добавить ELMo в контролируемую модель, мы сначала заморозим веса biLM, а затем соединим вектор ELMo $ELMo_{task\ k}$ с x_k и передаем расширенное представление ELMo $[x_k; ELMo_{task\ k}]$ в задачу RNN. Для некоторых задач (например, SNLI, SQuAD) мы наблюдаем дальнейшие улучшения за счет включения ELMo на выходе задачи RNN путем введения другого набора выходных удельных линейных весов и замены h_k на $[h_k; ELMo_{task\ k}]$. Поскольку остальная часть контролируемой модели остается неизменной, эти добавления могут происходить в контексте более сложных нейронных моделей. Например, см. эксперименты с SNLI в разд. 4, где уровень двойного внимания следует за biLSTM, или эксперименты с разрешением кореферентности, где модель кластеризации накладывается поверх biLSTM.
Finally, we found it beneficial to add a moderate amount of dropout to ELMo (Srivastava et al., 2014) and in some cases to regularize the ELMo weights by adding $\lambda_k w_k$ to the loss. This imposes an inductive bias on the ELMo weights to stay close to an average of all biLM layers.	Наконец, мы обнаружили, что полезно добавить умеренное количество отсева к ELMo (Srivastava et al., 2014) и в некоторых случаях упорядочить веса ELMo, добавив к потерям $\lambda_k w_k$. Это накладывает индуктивное смещение на веса ELMo, чтобы они оставались близкими к среднему значению всех слоев biLM.
3.4 Pre-trained bidirectional language model Architecture	3.4 Архитектура предварительно обученной двунаправленной языковой модели
The pre-trained biLMs in this paper are similar to the architectures in J'ozefowicz et al. (2016) and Kim et al. (2015), but modified to support joint	Предварительно обученные biLM в этой статье аналогичны архитектурам J'ozefowicz et al. (2016) и Ким и соавт. (2015), но изменен

training of both directions and add a residual connection between LSTM layers. We focus on large scale biLMs in this work, as Peters et al. (2017) highlighted the importance of using biLMs over forward-only LMs and large scale training.	для поддержки совместного обучения обоих направлений и добавления остаточной связи между слоями LSTM. В этой работе мы фокусируемся на крупномасштабных биЛМ, так как Peters et al. (2017) подчеркнули важность использования биЛМ по сравнению с прямыми ЛМ и крупномасштабным обучением.
To balance overall language model perplexity with model size and computational requirements for downstream tasks while maintaining a purely character-based input representation, we halved all embedding and hidden dimensions from the single best model CNN-BIG-LSTM in J'ozefowicz et al. (2016). The final model uses $L = 2$ biLSTM layers with 4096 units and 512 dimension projections and a residual connection from the first to second layer. The context insensitive type representation uses 2048 character n-gram convolutional filters followed by two highway layers (Srivastava et al., 2015) and a linear projection down to a 512 representation. As a result, the biLM provides three layers of representations for each input token, including those outside the training set due to the purely character input. In contrast, traditional word embedding methods only provide one layer of representation for tokens in a fixed vocabulary.	Чтобы сбалансировать общую сложность языковой модели с размером модели и вычислительными требованиями для последующих задач, сохраняя при этом чисто символьное представление ввода, мы вдвое сократили все встраиваемые и скрытые измерения из единственной лучшей модели CNN-BIG-LSTM в J'ozefowicz et al. (2016). В окончательной модели используются $L = 2$ слоя biLSTM с 4096 единицами и 512 размерными проекциями, а также остаточная связь между первым и вторым слоями. Контекстно-независимое представление типа использует 2048-символьные сверточные фильтры n-грамм, за которыми следуют два слоя шоссе (Srivastava et al., 2015) и линейная проекция до 512-представления. В результате биЛМ обеспечивает три уровня представлений для каждого входного токена, включая те, которые находятся за пределами обучающего набора из-за чисто символьного ввода. Напротив, традиционные методы встраивания слов обеспечивают только один уровень представления токенов в фиксированном словаре.
After training for 10 epochs on the 1B Word Benchmark (Chelba et al., 2014), the average forward and backward perplexities is 39.7, compared to 30.0 for the forward CNN-BIG-LSTM. Generally, we found the forward and backward perplexities to be approximately equal, with the backward value slightly lower.	После обучения в течение 10 эпох на тесте 1B Word Benchmark (Челба и др., 2014) среднее недоумение вперед и назад составляет 39,7 по сравнению с 30,0 для прямого CNN-BIG-LSTM. Как правило, мы обнаружили, что затруднения в прямом и обратном направлении примерно равны, а значение в обратном направлении немного ниже.
Once pretrained, the biLM can compute representations for any task. In some cases, fine tuning the biLM on domain specific data leads to significant drops in perplexity and an increase in downstream task performance. This can be seen as a type of domain transfer for the biLM. As a result, in most cases we used a fine-tuned biLM in the downstream task. See supplemental material for details.	После предварительной подготовки биЛМ может вычислять представления для любой задачи. В некоторых случаях точная настройка биЛМ на данных, специфичных для предметной области, приводит к значительному снижению сложности и увеличению производительности последующих задач. Это можно рассматривать как тип передачи домена для биЛМ. В результате в большинстве случаев мы использовали доработанный биЛМ в нисходящей задаче. Подробности смотрите в дополнительных материалах.
4 Evaluation	4 Оценка
Table 1 shows the performance of ELMo across a diverse set of six benchmark NLP tasks. In every task considered, simply adding ELMo establishes a new state-of-the-art result, with relative error reductions ranging from 6 - 20% over	В таблице 1 показана производительность ELMo в разнообразном наборе из шести эталонных задач НЛП. В каждой рассматриваемой задаче простое добавление ELMo обеспечивает новый современный результат с

strong base models. This is a very general result across a diverse set model architectures and language understanding tasks. In the remainder of this section we provide high-level sketches of the individual task results; see the supplemental material for full experimental details.

Question answering The Stanford Question Answering Dataset (SQuAD) (Rajpurkar et al., 2016) contains 100K+ crowd sourced questionanswer pairs where the answer is a span in a given Wikipedia paragraph. Our baseline model (Clark and Gardner, 2017) is an improved version of the Bidirectional Attention Flow model in Seo et al. (BiDAF; 2017). It adds a self-attention layer after the bidirectional attention component, simplifies some of the pooling operations and substitutes the LSTMs for gated recurrent units (GRUs; Cho et al., 2014). After adding ELMo to the baseline model, test set F1 improved by 4.7% from 81.1% to 85.8%, a 24.9% relative error reduction over the baseline, and improving the overall single model state-of-the-art by 1.4%. A 11 member ensemble pushes F1 to 87.4, the overall state-of-the-art at time of submission to the leaderboard.² The increase of 4.7% with ELMo is also significantly larger than the 1.8% improvement from adding CoVe to a baseline model (McCann et al., 2017).

относительным снижением погрешности в диапазоне от 6 до 20% по сравнению с сильными базовыми моделями. Это очень общий результат для различных архитектур моделей набора и задач понимания языка. В оставшейся части этого раздела мы приводим общие наброски результатов отдельных задач; см. дополнительный материал для получения полной экспериментальной информации.

Ответы на вопросы Стэнфордский набор данных для ответов на вопросы (SQuAD) (Rajpurkar et al., 2016) содержит более 100 000 пар вопросов и ответов, полученных из краудсорсинга, где ответ представляет собой интервал в заданном абзаце Википедии. Наша базовая модель (Clark and Gardner, 2017) представляет собой улучшенную версию модели двунаправленного потока внимания Seo et al. (БиДАФ; 2017). Он добавляет уровень самоконтроля после компонента двунаправленного внимания, упрощает некоторые операции объединения и заменяет LSTM для закрытых рекуррентных единиц (GRU; Cho et al., 2014). После добавления ELMo к базовой модели тестовый набор F1 улучшился на 4,7 % с 81,1 % до 85,8 %, относительное снижение погрешности на 24,9 % по сравнению с базовым уровнем и улучшение общего состояния отдельной модели на 1,4 %. Ансамбль из 11 участников поднимает F1 до 87,4, что является общим состоянием дел на момент подачи заявки в таблицу лидеров.² Увеличение на 4,7% с ELMo также значительно больше, чем улучшение на 1,8% от добавления CoVe к базовой модели. (Макканн и др., 2017).

TASK	PREVIOUS SOTA		OUR BASELINE	ELMo + BASELINE	INCREASE (ABSOLUTE/ RELATIVE)
SQuAD	Liu et al. (2017)	84.4	81.1	85.8	4.7 / 24.9%
SNLI	Chen et al. (2017)	88.6	88.0	88.7 ± 0.17	0.7 / 5.8%
SRL	He et al. (2017)	81.7	81.4	84.6	3.2 / 17.2%
Coref	Lee et al. (2017)	67.2	67.2	70.4	3.2 / 9.8%
NER	Peters et al. (2017)	91.93 ± 0.19	90.15	92.22 ± 0.10	2.06 / 21%
SST-5	McCann et al. (2017)	53.7	51.4	54.7 ± 0.5	3.3 / 6.8%

Table 1: Test set comparison of ELMo enhanced neural models with state-of-the-art single model baselines across six benchmark NLP tasks. The performance metric varies across tasks – accuracy for SNLI and SST-5; F1 for SQuAD, SRL and NER; average F1 for Coref. Due to the small test sizes for NER and SST-5, we report the mean and standard deviation across five runs with different random seeds. The “increase” column lists both the absolute and

Таблица 1. Сравнение набора тестов улучшенных нейронных моделей ELMo с современными базовыми уровнями одной модели в шести эталонных задачах НЛП. Метрика производительности варьируется в зависимости от задач — точность для SNLI и SST-5; F1 для SQuAD, SRL и NER; средний F1 для Coref. Из-за небольших размеров тестов для NER и SST-5 мы сообщаем среднее значение и стандартное отклонение для пяти прогонов с разными случайными начальными значениями. В столбце «увеличение» перечислены как абсолютные, так и относительные улучшения по сравнению с нашим базовым уровнем.

relative improvements over our baseline.	
<p>Textual entailment Textual entailment is the task of determining whether a “hypothesis” is true, given a “premise”. The Stanford Natural Language Inference (SNLI) corpus (Bowman et al., 2015) provides approximately 550K hypothesis/premise pairs. Our baseline, the ESIM sequence model from Chen et al. (2017), uses a biLSTM to encode the premise and hypothesis, followed by a matrix attention layer, a local inference layer, another biLSTM inference composition layer, and finally a pooling operation before the output layer. Overall, adding ELMo to the ESIM model improves accuracy by an average of 0.7% across five random seeds. A five member ensemble pushes the overall accuracy to 89.3%, exceeding the previous ensemble best of 88.9% (Gong et al., 2018).</p>	<p>Текстовое следствие Текстовое следствие — это задача определения того, верна ли «гипотеза» при наличии «предпосылки». Корпус Stanford Natural Language Inference (SNLI) (Bowman et al., 2015) содержит около 550 000 пар гипотез/посылок. Наш базовый уровень, модель последовательности ESIM от Chen et al. (2017) использует biLSTM для кодирования предпосылки и гипотезы, за которым следует уровень матричного внимания, уровень локального вывода, еще один уровень композиции вывода biLSTM и, наконец, операция объединения перед выходным уровнем. В целом, добавление ELMo к модели ESIM повышает точность в среднем на 0,7% для пяти случайных исходных значений. Ансамбль из пяти членов увеличивает общую точность до 89,3%, превысив предыдущий лучший результат ансамбля в 88,9% (Gong et al., 2018).</p>
<p>Semantic role labeling A semantic role labeling (SRL) system models the predicate-argument structure of a sentence, and is often described as answering “Who did what to whom”. He et al. (2017) modeled SRL as a BIO tagging problem and used an 8-layer deep biLSTM with forward and backward directions interleaved, following Zhou and Xu (2015). As shown in Table 1, when adding ELMo to a re-implementation of He et al. (2017) the single model test set F1 jumped 3.2% from 81.4% to 84.6% – a new state-of-the-art on the OntoNotes benchmark (Pradhan et al., 2013), even improving over the previous best ensemble result by 1.2%.</p>	<p>Семантическая ролевая маркировка Система семантической ролевой маркировки (SRL) моделирует структуру предиката-аргумента предложения и часто описывается как ответ на вопрос «кто что кому сделал». Он и др. (2017) смоделировали SRL как проблему маркировки BIO и использовали 8-слойный глубокий biLSTM с чередованием прямых и обратных направлений, следуя Zhou и Xu (2015). Как показано в таблице 1, при добавлении ELMo к повторной реализации He et al. (2017) тестовый набор одной модели F1 подскочил на 3,2 % с 81,4 % до 84,6 % — новый современный показатель в тесте OntoNotes (Pradhan et al., 2013), даже улучшив предыдущий лучший результат ансамбля на 1,2%.</p>
<p>Coreference resolution Coreference resolution is the task of clustering mentions in text that refer to the same underlying real world entities. Our baseline model is the end-to-end span-based neural model of Lee et al. (2017). It uses a biLSTM and attention mechanism to first compute span representations and then applies a softmax mention ranking model to find coreference chains. In our experiments with the OntoNotes coreference annotations from the CoNLL 2012 shared task (Pradhan et al., 2012), adding ELMo improved the average F1 by 3.2% from 67.2 to 70.4, establishing a new state of the art, again improving over the previous best ensemble result by 1.6% F1.</p>	<p>Разрешение кореференции Разрешение кореференции — это задача кластеризации упоминаний в тексте, которые относятся к одним и тем же базовым объектам реального мира. Наша базовая модель — это сквозная нейронная модель Lee et al. (2017). Он использует biLSTM и механизм внимания, чтобы сначала вычислить представления диапазона, а затем применяет модель ранжирования упоминаний softmax для поиска цепочек кореферентности. В наших экспериментах с аннотациями кореференции OntoNotes из общей задачи CoNLL 2012 (Pradhan et al., 2012) добавление ELMo улучшило средний F1 на 3,2% с 67,2 до 70,4, установив новый уровень техники, снова улучшив предыдущий. лучший ансамблевый результат на 1,6% F1.</p>
Named entity extraction The CoNLL 2003 NER task (Sang and Meulder,	Извлечение именованных сущностей Задача CoNLL 2003 NER (Sang and

2003) consists of newswire from the Reuters RCV1 corpus tagged with four different entity types (PER, LOC, ORG, MISC). Following recent state-of-the-art systems (Lample et al., 2016; Peters et al., 2017), the baseline model uses pre-trained word embeddings, a character-based CNN representation, two biLSTM layers and a conditional random field (CRF) loss (Lafferty et al., 2001), similar to Collobert et al. (2011). As shown in Table 1, our ELMo enhanced biLSTM-CRF achieves 92.22% F1 averaged over five runs. The key difference between our system and the previous state of the art from Peters et al. (2017) is that we allowed the task model to learn a weighted average of all biLM layers, whereas Peters et al. (2017) only use the top biLM layer. As shown in Sec. 5.1, using all layers instead of just the last layer improves performance across multiple tasks.

Meulder, 2003) состоит из ленты новостей из корпуса Reuters RCV1, помеченной четырьмя различными типами сущностей (PER, LOC, ORG, MISC). Следуя новейшим современным системам (Lample et al., 2016; Peters et al., 2017), базовая модель использует предварительно обученные вложения слов, символьное представление CNN, два слоя biLSTM и условную случайную выборку. потеря поля (CRF) (Lafferty et al., 2001), аналогично Collobert et al. (2011). Как показано в Таблице 1, наш улучшенный ELMo biLSTM-CRF достигает 92,22% F1, усредненного по пяти запускам. Ключевое отличие нашей системы от предыдущего уровня техники от Peters et al. (2017), заключается в том, что мы позволили модели задачи изучить средневзвешенное значение всех слоев biLM, тогда как Peters et al. (2017) используют только верхний слой biLM. Как показано в разд. 5.1 использование всех слоев, а не только последнего слоя, повышает производительность при выполнении нескольких задач.

Sentiment analysis The fine-grained sentiment classification task in the Stanford Sentiment Treebank (SST-5; Socher et al., 2013) involves selecting one of five labels (from very negative to very positive) to describe a sentence from a movie review. The sentences contain diverse linguistic phenomena such as idioms and complex syntactic constructions such as negations that are difficult for models to learn. Our baseline model is the biattentive classification network (BCN) from McCann et al. (2017), which also held the prior state-of-the-art result when augmented with CoVe embeddings. Replacing CoVe with ELMo in the BCN model results in a 1.0% absolute accuracy improvement over the state of the art.

Анализ настроений Задача детальной классификации настроений в Stanford Sentiment Treebank (SST-5; Socher et al., 2013) включает выбор одной из пяти меток (от очень негативных до очень позитивных) для описания предложения из рецензии на фильм. Предложения содержат разнообразные лингвистические явления, такие как идиомы, и сложные синтаксические конструкции, такие как отрицания, которые трудно усвоить моделям. Нашей базовой моделью является сеть двухсторонней классификации (BCN) от McCann et al. (2017), который также содержал предыдущий современный результат при дополнении вложениями CoVe. Замена CoVe на ELMo в модели BCN приводит к повышению абсолютной точности на 1,0% по сравнению с современным уровнем техники.

Task	Baseline	Last Only	All layers	
			$\lambda=1$	$\lambda=0.001$
SQuAD	80.8	84.7	85.0	85.2
SNLI	88.1	89.1	89.3	89.5
SRL	81.6	84.1	84.6	84.8

Таблица 2. Производительность набора для разработки для SQuAD, SNLI и SRL в сравнении с использованием всех слоев biLM (с различными вариантами силы регуляризации λ) только с верхним слоем.

Table 2: Development set performance for SQuAD, SNLI and SRL

comparing using all layers of the biLM (with different choices of regularization strength λ) to just the top layer.				Таблица 3: Производительность набора для разработки для SQuAD, SNLI и SRL при включении ELMo в разных местах в контролируемой модели.
Task	Input Only	Input & Output	Output Only	
SQuAD	85.1	85.6	84.8	
SNLI	88.9	89.5	88.7	
SRL	84.7	84.3	80.9	
Table 3: Development set performance for SQuAD, SNLI and SRL when including ELMo at different locations in the supervised model.				
5 Analysis				5 Анализ
This section provides an ablation analysis to validate our chief claims and to elucidate some interesting aspects of ELMo representations. Sec. 5.1 shows that using deep contextual representations in downstream tasks improves performance over previous work that uses just the top layer, regardless of whether they are produced from a biLM or MT encoder, and that ELMo representations provide the best overall performance. Sec. 5.3 explores the different types of contextual information captured in biLMs and uses two intrinsic evaluations to show that syntactic information is better represented at lower layers while semantic information is captured a higher layers, consistent with MT encoders. It also shows that our biLM consistently provides richer representations than CoVe. Additionally, we analyze the sensitivity to where ELMo is included in the task model (Sec. 5.2), training set size (Sec. 5.4), and visualize the ELMo learned weights across the tasks (Sec. 5.5).				В этом разделе представлен анализ абляции, чтобы подтвердить наши основные утверждения и прояснить некоторые интересные аспекты представлений ELMo. сек. 5.1 показано, что использование глубоких контекстных представлений в последующих задачах повышает производительность по сравнению с предыдущей работой, в которой используется только верхний уровень, независимо от того, созданы ли они кодером biLM или MT, и что представления ELMo обеспечивают наилучшую общую производительность. сек. 5.3 исследует различные типы контекстной информации, захваченной в biLM, и использует две внутренние оценки, чтобы показать, что синтаксическая информация лучше представлена на нижних уровнях, в то время как семантическая информация захвачена на более высоких уровнях, что согласуется с кодировщиками MT. Это также показывает, что наш biLM постоянно обеспечивает более богатые представления, чем CoVe. Кроме того, мы анализируем чувствительность к тому, где ELMo включен в модель задачи (раздел 5.2), размер обучающей выборки (раздел 5.4) и визуализируем изученные веса ELMo для задач (раздел 5.5).
5.1 Alternate layer weighting schemes				5.1 Альтернативные схемы взвешивания слоев
There are many alternatives to Equation 1 for combining the biLM layers. Previous work on contextual representations used only the last layer, whether it be from a biLM (Peters et al., 2017) or an MT encoder (CoVe; McCann et al., 2017). The choice of the regularization parameter λ is also important, as large values such as $\lambda = 1$ effectively reduce the weighting function to a				Существует множество альтернатив уравнению 1 для объединения слоев biLM. Предыдущая работа над контекстными представлениями использовала только последний слой, будь то кодировщик biLM (Peters et al., 2017) или MT (CoVe; McCann et al., 2017). Выбор параметра регуляризации λ также важен, так как большие значения, такие как $\lambda = 1$,

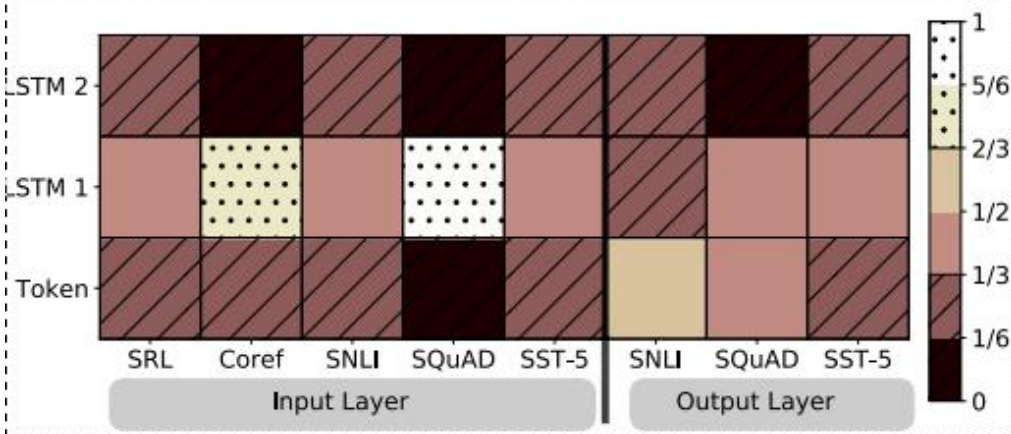
simple average over the layers, while smaller values (e.g., $\lambda = 0.001$) allow the layer weights to vary.	эффективно сводят весовую функцию к простому среднему по слоям, в то время как меньшие значения (например, $\lambda = 0,001$) позволяют варьировать веса слоев.
Table 2 compares these alternatives for SQuAD, SNLI and SRL. Including representations from all layers improves overall performance over just using the last layer, and including contextual representations from the last layer improves performance over the baseline. For example, in the case of SQuAD, using just the last biLM layer improves development F1 by 3.9% over the baseline. Averaging all biLM layers instead of using just the last layer improves F1 another 0.3% (comparing “Last Only” to $\lambda=1$ columns), and allowing the task model to learn individual layer weights improves F1 another 0.2% ($\lambda=1$ vs. $\lambda=0.001$). A small λ is preferred in most cases with ELMo, although for NER, a task with a smaller training set, the results are insensitive to λ (not shown).	В таблице 2 сравниваются эти альтернативы для SQuAD, SNLI и SRL. Включение представлений со всех уровней повышает общую производительность по сравнению с использованием только последнего уровня, а включение контекстных представлений из последнего уровня повышает производительность по сравнению с базовым уровнем. Например, в случае SQuAD использование только последнего слоя biLM улучшает разработку F1 на 3,9% по сравнению с базовым уровнем. Усреднение всех слоев biLM вместо использования только последнего слоя улучшает F1 еще на 0,3% (по сравнению с параметром «Только последний» для $\lambda=1$ столбцов), а разрешение модели задачи изучать веса отдельных слоев улучшает F1 еще на 0,2% ($\lambda=1$ против 1 столбца). $\lambda=0,001$). В большинстве случаев с ELMo предпочтительнее небольшое значение λ , хотя для NER, задачи с меньшим набором обучающих данных, результаты нечувствительны к λ (не показано).
The overall trend is similar with CoVe but with smaller increases over the baseline. For SNLI, averaging all layers with $\lambda=1$ improves development accuracy from 88.2 to 88.7% over using just the last layer. SRL F1 increased a marginal 0.1% to 82.2 for the $\lambda=1$ case compared to using the last layer only.	Общая тенденция аналогична CoVe, но с меньшим увеличением по сравнению с исходным уровнем. Для SNLI усреднение всех слоев с $\lambda=1$ повышает точность разработки с 88,2 до 88,7% по сравнению с использованием только последнего слоя. SRL F1 увеличился на маргинальные 0,1% до 82,2 для случая $\lambda=1$ по сравнению с использованием только последнего слоя.
5.2 Where to include ELMo?	5.2 Куда включать ELMo?
All of the task architectures in this paper include word embeddings only as input to the lowest layer biRNN. However, we find that including ELMo at the output of the biRNN in task-specific architectures improves overall results for some tasks. As shown in Table 3, including ELMo at both the input and output layers for SNLI and SQuAD improves over just the input layer, but for SRL (and coreference resolution, not shown) performance is highest when it is included at just the input layer. One possible explanation for this result is that both the SNLI and SQuAD architectures use attention layers after the biRNN, so introducing ELMo at this layer allows the model to attend directly to the biLM’s internal representations. In the SRL case, the task-specific context representations are likely more important than those from the biLM.	Все архитектуры задач в этой статье включают встраивание слов только в качестве входных данных для нижнего уровня biRNN. Однако мы обнаружили, что включение ELMo на выходе biRNN в архитектурах для конкретных задач улучшает общие результаты для некоторых задач. Как показано в Таблице 3, включение ELMo как на входном, так и на выходном уровнях для SNLI и SQuAD улучшается только на входном уровне, но для SRL (и разрешения кореференса, не показано) производительность является самой высокой, когда он включен только на входном уровне. Одно из возможных объяснений этого результата заключается в том, что и в архитектуре SNLI, и в архитектуре SQuAD уровни внимания используются после biRNN, поэтому введение ELMo на этом уровне позволяет модели напрямую обращаться к внутренним представлениям biLM. В случае SRL представления контекста для

		конкретной задачи, вероятно, более важны, чем представления из biLM.																
<table><tr><th>Source</th><th>Nearest Neighbors</th></tr><tr><td>GloVe play</td><td>playing, game, games, played, players, plays, player, Play, football, multiplayer</td></tr><tr><td rowspan="2">biLM</td><td>Chico Ruiz made a spectacular <u>play</u> on Alusik 's grounder {...}</td></tr><tr><td>Olivia De Havilland signed to do a Broadway <u>play</u> for Garson {...}</td></tr></table>		Source	Nearest Neighbors	GloVe play	playing, game, games, played, players, plays, player, Play, football, multiplayer	biLM	Chico Ruiz made a spectacular <u>play</u> on Alusik 's grounder {...}	Olivia De Havilland signed to do a Broadway <u>play</u> for Garson {...}	Таблица 4: Ближайшие соседи для «игры» с использованием GloVe и вложений контекста из biLM.									
Source	Nearest Neighbors																	
GloVe play	playing, game, games, played, players, plays, player, Play, football, multiplayer																	
biLM	Chico Ruiz made a spectacular <u>play</u> on Alusik 's grounder {...}																	
	Olivia De Havilland signed to do a Broadway <u>play</u> for Garson {...}																	
Table 4: Nearest neighbors to “play” using GloVe and the context embeddings from a biLM.																		
<table><tr><th>Model</th><th>F₁</th></tr><tr><td>WordNet 1st Sense Baseline</td><td>65.9</td></tr><tr><td>Raganato et al. (2017a)</td><td>69.9</td></tr><tr><td>Iacobacci et al. (2016)</td><td>70.1</td></tr><tr><td>CoVe, First Layer</td><td>59.4</td></tr><tr><td>CoVe, Second Layer</td><td>64.7</td></tr><tr><td>biLM, First layer</td><td>67.4</td></tr><tr><td>biLM, Second layer</td><td>69.0</td></tr></table>		Model	F ₁	WordNet 1st Sense Baseline	65.9	Raganato et al. (2017a)	69.9	Iacobacci et al. (2016)	70.1	CoVe, First Layer	59.4	CoVe, Second Layer	64.7	biLM, First layer	67.4	biLM, Second layer	69.0	Таблица 5: Мелкозернистый WSD F1 для всех слов. Для CoVe и biLM мы сообщаем оценки как для первого, так и для второго уровня biLSTM.
Model	F ₁																	
WordNet 1st Sense Baseline	65.9																	
Raganato et al. (2017a)	69.9																	
Iacobacci et al. (2016)	70.1																	
CoVe, First Layer	59.4																	
CoVe, Second Layer	64.7																	
biLM, First layer	67.4																	
biLM, Second layer	69.0																	
Table 5: All-words fine grained WSD F1. For CoVe and the biLM, we report scores for both the first and second layer biLSTMs.																		

Model	Acc.	Таблица 6: Тестовый набор Точность маркировки POS для PTB. Для CoVe и biLM мы сообщаем оценки как для первого, так и для второго уровня biLSTM.
Collobert et al. (2011)	97.3	
Ma and Hovy (2016)	97.6	
Ling et al. (2015)	97.8	
CoVe, First Layer	93.3	
CoVe, Second Layer	92.8	
biLM, First Layer	97.3	
biLM, Second Layer	96.8	
Table 6: Test set POS tagging accuracies for PTB. For CoVe and the biLM, we report scores for both the first and second layer biLSTMs.		
5.3 What information is captured by the biLM’s representations?		5.3 Какая информация содержится в представлениях biLM?
<p>Since adding ELMo improves task performance over word vectors alone, the biLM’s contextual representations must encode information generally useful for NLP tasks that is not captured in word vectors. Intuitively, the biLM must be disambiguating the meaning of words using their context. Consider “play”, a highly polysemous word. The top of Table 4 lists nearest neighbors to “play” using GloVe vectors. They are spread across several parts of speech (e.g., “played”, “playing” as verbs, and “player”, “game” as nouns) but concentrated in the sportsrelated senses of “play”. In contrast, the bottom two rows show nearest neighbor sentences from the SemCor dataset (see below) using the biLM’s context representation of “play” in the source sentence. In these cases, the biLM is able to disambiguate both the part of speech and word sense in the source sentence.</p>		<p>Поскольку добавление ELMo повышает производительность задачи по сравнению с одними только векторами слов, контекстуальные представления biLM должны кодировать информацию, обычно полезную для задач NLP, которая не фиксируется в векторах слов. Интуитивно понятно, что biLM должен устранять неоднозначность значений слов, используя их контекст. Возьмем слово «играть» — очень многозначное слово. В верхней части таблицы 4 перечислены ближайшие соседи для «игры» с использованием векторов GloVe. Они распределены по нескольким частям речи (например, «играл», «играл» как глаголы и «игрок», «игра» как существительные), но сконцентрированы в связанных со спортом значениях слова «играть». Напротив, в двух нижних строках показаны предложения с ближайшими соседями из набора данных SemCor (см. ниже) с использованием контекстного представления biLM слова «играть» в исходном предложении. В этих случаях biLM может устранить неоднозначность как части речи, так и смысла слова в исходном предложении.</p>
<p>These observations can be quantified using an intrinsic evaluation of the contextual representations similar to Belinkov et al. (2017). To isolate the information encoded by the biLM, the representations are used to directly make predictions for a fine grained word sense disambiguation (WSD) task</p>		<p>Эти наблюдения могут быть количественно оценены с помощью внутренней оценки контекстуальных представлений, аналогичной Belinkov et al. (2017). Чтобы изолировать информацию, закодированную biLM, представления используются для прямого прогнозирования для</p>

<p>and a POS tagging task. Using this approach, it is also possible to compare to CoVe, and across each of the individual layers.</p>	<p>задачи устранения многозначности по смыслу (WSD) и задачи маркировки POS. Используя этот подход, также можно сравнивать с CoVe и по каждому из отдельных слоев.</p>
<p>Word sense disambiguation Given a sentence, we can use the biLM representations to predict the sense of a target word using a simple 1- nearest neighbor approach, similar to Melamud et al. (2016). To do so, we first use the biLM to compute representations for all words in SemCor 3.0, our training corpus (Miller et al., 1994), and then take the average representation for each sense. At test time, we again use the biLM to compute representations for a given target word and take the nearest neighbor sense from the training set, falling back to the first sense from WordNet for lemmas not observed during training.</p>	<p>Устранение неоднозначности смысла слова Для заданного предложения мы можем использовать представления biLM для предсказания смысла целевого слова, используя простой подход с 1 ближайшим соседом, аналогичный Меламуд и др. (2016). Для этого мы сначала используем biLM для вычисления представлений всех слов в SemCor 3.0, нашем обучающем корпусе (Miller et al., 1994), а затем берем среднее представление для каждого смысла. Во время тестирования мы снова используем biLM для вычисления представлений для заданного целевого слова и берем ближайший соседний смысл из обучающего набора, возвращаясь к первому смыслу из WordNet для лемм, не наблюдаемых во время обучения.</p>
<p>Table 5 compares WSD results using the evaluation framework from Raganato et al. (2017b) across the same suite of four test sets in Raganato et al. (2017a). Overall, the biLM top layer representations have F1 of 69.0 and are better at WSD than the first layer. This is competitive with a state-of-the-art WSD-specific supervised model using hand crafted features (Iacobacci et al., 2016) and a task specific biLSTM that is also trained with auxiliary coarse-grained semantic labels and POS tags (Raganato et al., 2017a). The CoVe biLSTM layers follow a similar pattern to those from the biLM (higher overall performance at the second layer compared to the first); however, our biLM outperforms the CoVe biLSTM, which trails the WordNet first sense baseline.</p>	<p>В таблице 5 сравниваются результаты WSD с использованием системы оценки Raganato et al. (2017b) в том же наборе из четырех наборов тестов в Raganato et al. (2017a). В целом, представления верхнего уровня biLM имеют F1 69,0 и лучше в WSD, чем первый уровень. Это конкурентоспособно с современной контролируемой моделью WSD, использующей созданные вручную функции (Iacobacci et al., 2016), и biLSTM для конкретной задачи, которая также обучается с помощью вспомогательных крупнозернистых семантических меток и тегов POS (Raganato et al., 2017a). Уровни CoVe biLSTM аналогичны слоям biLM (более высокая общая производительность на втором уровне по сравнению с первым); однако наш biLM превосходит CoVe biLSTM, который отстает от базового уровня первого чувства WordNet.</p>
<p>POS tagging To examine whether the biLM captures basic syntax, we used the context representations as input to a linear classifier that predicts POS tags with the Wall Street Journal portion of the Penn Treebank (PTB) (Marcus et al., 1993). As the linear classifier adds only a small amount of model capacity, this is direct test of the biLM’s representations. Similar to WSD, the biLM representations are competitive with carefully tuned, task specific biLSTMs (Ling et al., 2015; Ma and Hovy, 2016). However, unlike WSD, accuracies using the first biLM layer are higher than the top layer, consistent with results from deep biLSTMs in multi-task training (Søgaard and Goldberg, 2016; Hashimoto et al., 2017) and MT (Belinkov et al., 2017). CoVe POS tagging accuracies follow the same pattern as those from the biLM, and just like for</p>	<p>Маркировка POS Чтобы проверить, фиксирует ли biLM базовый синтаксис, мы использовали представления контекста в качестве входных данных для линейного классификатора, который предсказывает теги POS с частью Wall Street Journal в Penn Treebank (PTB) (Marcus et al., 1993). Поскольку линейный классификатор добавляет лишь небольшую емкость модели, это прямая проверка представлений biLM. Подобно WSD, представления biLM конкурируют с тщательно настроенными biLSTM для конкретных задач (Ling et al., 2015; Ma and Hovy, 2016). Однако, в отличие от WSD, точность при использовании первого слоя biLM выше, чем при использовании верхнего слоя, что согласуется с результатами глубоких biLSTM при многозадачном обучении (Søgaard</p>

WSD, the biLM achieves higher accuracies than the CoVe encoder.	and Goldberg, 2016; Hashimoto et al., 2017) и машинного обучения (Belinkov et al. , 2017). Точность маркировки CoVe POS соответствует той же схеме, что и у biLM, и, как и для WSD, biLM обеспечивает более высокую точность, чем кодировщик CoVe.																																								
Implications for supervised tasks Taken together, these experiments confirm different layers in the biLM represent different types of information and explain why including all biLM layers is important for the highest performance in downstream tasks. In addition, the biLM’s representations are more transferable to WSD and POS tagging than those in CoVe, helping to illustrate why ELMo outperforms CoVe in downstream tasks.	Последствия для контролируемых задач В совокупности эти эксперименты подтверждают, что разные уровни в biLM представляют разные типы информации, и объясняют, почему включение всех слоев biLM важно для максимальной производительности в последующих задачах. Кроме того, представления biLM лучше переносятся в теги WSD и POS, чем в CoVe, что помогает проиллюстрировать, почему ELMo превосходит CoVe в последующих задачах.																																								
5.4 Sample efficiency	5.4 Эффективность образца																																								
Adding ELMo to a model increases the sample efficiency considerably, both in terms of number of parameter updates to reach state-of-the-art performance and the overall training set size. For example, the SRL model reaches a maximum development F1 after 486 epochs of training without ELMo. After adding ELMo, the model exceeds the baseline maximum at epoch 10, a 98% relative decrease in the number of updates needed to reach the same level of performance.	Добавление ELMo в модель значительно повышает эффективность выборки, как с точки зрения количества обновлений параметров для достижения самой современной производительности, так и с точки зрения общего размера обучающей выборки. Например, модель SRL достигает максимального развития F1 после 486 эпох обучения без ELMo. После добавления ELMo модель превышает базовый максимум в эпоху 10, что означает относительное уменьшение количества обновлений на 98%, необходимых для достижения того же уровня производительности.																																								
<div><div><div>SNLI (Accuracy)</div><table><thead><tr><th>Training Set Size</th><th>With ELMo</th><th>Baseline</th><th>Gain</th></tr></thead><tbody><tr><td>0.1%</td><td>~61</td><td>~49</td><td>+12.3</td></tr><tr><td>1%</td><td>~75</td><td>~67</td><td>+7.7</td></tr><tr><td>10%</td><td>~85</td><td>~83</td><td>+1.5</td></tr><tr><td>100%</td><td>~89</td><td>~88</td><td>+1.4</td></tr></tbody></table></div><div><div>SRL (F1)</div><table><thead><tr><th>Training Set Size</th><th>With ELMo</th><th>Baseline</th><th>Gain</th></tr></thead><tbody><tr><td>0.1%</td><td>~37</td><td>~19</td><td>+18.6</td></tr><tr><td>1%</td><td>~65</td><td>~44</td><td>+20.5</td></tr><tr><td>10%</td><td>~77</td><td>~66</td><td>+10.8</td></tr><tr><td>100%</td><td>~85</td><td>~82</td><td>+3.1</td></tr></tbody></table></div></div>	Training Set Size	With ELMo	Baseline	Gain	0.1%	~61	~49	+12.3	1%	~75	~67	+7.7	10%	~85	~83	+1.5	100%	~89	~88	+1.4	Training Set Size	With ELMo	Baseline	Gain	0.1%	~37	~19	+18.6	1%	~65	~44	+20.5	10%	~77	~66	+10.8	100%	~85	~82	+3.1	Рисунок 1: Сравнение базовой производительности и производительности ELMo для SNLI и SRL при изменении размера обучающей выборки от 0,1% до 100%.
Training Set Size	With ELMo	Baseline	Gain																																						
0.1%	~61	~49	+12.3																																						
1%	~75	~67	+7.7																																						
10%	~85	~83	+1.5																																						
100%	~89	~88	+1.4																																						
Training Set Size	With ELMo	Baseline	Gain																																						
0.1%	~37	~19	+18.6																																						
1%	~65	~44	+20.5																																						
10%	~77	~66	+10.8																																						
100%	~85	~82	+3.1																																						

<p>Figure 1: Comparison of baseline vs. ELMo performance for SNLI and SRL as the training set size is varied from 0.1% to 100%.</p>	
 <p>Figure 2: Visualization of softmax normalized biLM layer weights across tasks and ELMo locations. Normalized weights less then 1/3 are hatched with horizontal lines and those greater then 2/3 are speckled.</p>	<p>Рисунок 2: Визуализация нормализованных весов слоев biLM softmax для задач и местоположений ELMo. Нормализованные веса менее 1/3 заштрихованы горизонтальными линиями, а веса больше 2/3 заштрихованы.</p>
<p>In addition, ELMo-enhanced models use smaller training sets more efficiently than models without ELMo. Figure 1 compares the performance of baselines models with and without ELMo as the percentage of the full training set is varied from 0.1% to 100%. Improvements with ELMo are largest for smaller training sets and significantly reduce the amount of training data needed to reach a given level of performance. In the SRL case, the ELMo model with 1% of the training set has about the same F1 as the baseline model with 10% of the training set.</p>	<p>Кроме того, модели с расширенными возможностями ELMo более эффективно используют меньшие обучающие наборы, чем модели без ELMo. На рис. 1 сравнивается производительность базовых моделей с ELMo и без него, поскольку процент полного обучающего набора варьируется от 0,1% до 100%. Улучшения с ELMo являются самыми большими для небольших обучающих наборов и значительно сокращают объем обучающих данных, необходимых для достижения заданного уровня производительности. В случае SRL модель ELMo с 1% обучающей выборки имеет примерно такой же F1, что и базовая модель с 10% обучающей выборки.</p>
<p>5.5 Visualization of learned weights</p>	<p>5.5 Визуализация изученных весов</p>
<p>Figure 2 visualizes the softmax-normalized learned layer weights. At the input layer, the task model favors the first biLSTM layer. For coreference and SQuAD, this is strongly favored, but the distribution is less peaked for the other tasks. The output layer weights are relatively balanced, with a slight preference for the lower layers.</p>	<p>На рис. 2 показаны весовые коэффициенты изученного слоя, нормализованные с помощью softmax. На входном уровне модель задачи отдает предпочтение первому уровню biLSTM. Для кореференса и SQuAD это сильно предпочтительнее, но для других задач распределение менее пиковое. Веса выходных слоев относительно сбалансированы, с небольшим предпочтением нижних слоев.</p>
<p>6 Conclusion</p>	<p>6. Заключение</p>
<p>We have introduced a general approach for learning high-quality deep context-dependent representations from biLMs, and shown large</p>	<p>Мы представили общий подход к изучению высококачественных глубоких контекстно-зависимых представлений из biLM и</p>

<p>improvements when applying ELMo to a broad range of NLP tasks. Through ablations and other controlled experiments, we have also confirmed that the biLM layers efficiently encode different types of syntactic and semantic information about words in context, and that using all layers improves overall task performance.</p>	<p>продемонстрировали значительные улучшения при применении ELMo к широкому кругу задач НЛП. С помощью абляции и других контролируемых экспериментов мы также подтвердили, что слои biLM эффективно кодируют различные типы синтаксической и семантической информации о словах в контексте, и что использование всех слоев повышает общую производительность задачи.</p>
<p>References</p>	
<p>Jimmy Ba, Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. CoRR abs/1607.06450.</p> <p>Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James R. Glass. 2017. What do neural machine translation models learn about morphology? In ACL.</p> <p>Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. TACL 5:135–146.</p> <p>Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics.</p> <p>Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. 2014. One billion word benchmark for measuring progress in statistical language modeling. In INTERSPEECH.</p> <p>Qian Chen, Xiao-Dan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced lstm for natural language inference. In ACL.</p> <p>Jason Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional LSTM-CNNs. In TACL.</p> <p>Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. In SSST@EMNLP.</p> <p>Christopher Clark and Matthew Gardner. 2017. Simple and effective multi-paragraph reading comprehension. CoRR abs/1710.10723.</p> <p>Kevin Clark and Christopher D. Manning. 2016. Deep</p>	

reinforcement learning for mention-ranking coreference models. In EMNLP.

Ronan Collobert, Jason Weston, L'eon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel P. Kuksa. 2011. Natural language processing (almost) from scratch. In JMLR.

Andrew M. Dai and Quoc V. Le. 2015. Semi-supervised sequence learning. In NIPS.

Greg Durrett and Dan Klein. 2013. Easy victories and uphill battles in coreference resolution. In EMNLP.

Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. In NIPS.

Yichen Gong, Heng Luo, and Jian Zhang. 2018. Natural language inference over interaction space. In ICLR.

Kazuma Hashimoto, Caiming Xiong, Yoshimasa Tsurukawa, and Richard Socher. 2017. A joint many-task model: Growing a neural network for multiple nlp tasks. In EMNLP 2017.

Luheng He, Kenton Lee, Mike Lewis, and Luke S. Zettlemoyer. 2017. Deep semantic role labeling: What works and what's next. In ACL.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9.

Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. 2016. Embeddings for word sense disambiguation: An evaluation study. In ACL.

Rafal J'ozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. 2016. Exploring the limits of language modeling. CoRR abs/1602.02410.

Rafal J'ozefowicz, Wojciech Zaremba, and Ilya Sutskever. 2015. An empirical exploration of recurrent network architectures. In ICML.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. 2015. Character-aware neural language models. In AAAI 2016.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In ICLR.

Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, Ishaan Gulrajani James Bradbury, Victor Zhong, Romain Paulus, and Richard Socher. 2016. Ask me anything: Dynamic memory networks for natural language processing. In ICML.

John D. Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In ICML.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In NAACL-HLT.

Kenton Lee, Luheng He, Mike Lewis, and Luke S. Zettlemoyer. 2017. End-to-end neural coreference resolution. In EMNLP.

Wang Ling, Chris Dyer, Alan W. Black, Isabel Trancoso, Ramon Fernandez, Silvio Amir, Luís Marujo, and Tiago Luís. 2015. Finding function in form: Compositional character models for open vocabulary word representation. In EMNLP.

Xiaodong Liu, Yelong Shen, Kevin Duh, and Jianfeng Gao. 2017. Stochastic answer networks for machine reading comprehension. arXiv preprint arXiv:1712.03556.

Xuezhe Ma and Eduard H. Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In ACL.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of english: The penn treebank. Computational Linguistics 19:313–330.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In NIPS 2017.

Oren Melamud, Jacob Goldberger, and Ido Dagan. 2016. context2vec: Learning generic context embedding with bidirectional

lstm. In CoNLL. Gábor Melis, Chris Dyer, and Phil Blunsom. 2017. On the state of the art of evaluation in neural language models. CoRR abs/1707.05589.

Stephen Merity, Nitish Shirish Keskar, and Richard Socher. 2017. Regularizing and optimizing lstm language models. CoRR abs/1708.02182.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In NIPS.

George A. Miller, Martin Chodorow, Shari Landes, Claudia Leacock, and Robert G. Thomas. 1994. Using a semantic concordance for sense identification. In HLT.

Tsendsuren Munkhdalai and Hong Yu. 2017. Neural tree indexers for text understanding. In EACL.

Arvind Neelakantan, Jeevan Shankar, Alexandre Passos, and Andrew McCallum. 2014. Efficient non parametric estimation of multiple embeddings per word in vector space. In EMNLP.

Martha Palmer, Paul Kingsbury, and Daniel Gildea. 2005. The proposition bank: An annotated corpus of semantic roles. Computational Linguistics 31:71–106.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In EMNLP.

Matthew E. Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. 2017. Semi-supervised sequence tagging with bidirectional language models. In ACL.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using ontonotes. In CoNLL.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. Conll-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes. In EMNLP CoNLL Shared Task.

<p>Alessandro Raganato, Claudio Delli Bovi, and Roberto Navigli. 2017a. Neural sequence learning models for word sense disambiguation. In EMNLP.</p> <p>Alessandro Raganato, Jose Camacho-Collados, and Roberto Navigli. 2017b. Word sense disambiguation: A unified evaluation framework and empirical comparison. In EACL.</p> <p>Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. In EMNLP.</p> <p>Prajit Ramachandran, Peter Liu, and Quoc Le. 2017. Improving sequence to sequence learning with unlabeled data. In EMNLP.</p> <p>Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In CoNLL.</p> <p>Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. In ICLR.</p> <p>Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In EMNLP.</p> <p>Anders Søgaard and Yoav Goldberg. 2016. Deep multi-task learning with low level tasks supervised at lower layers. In ACL 2016.</p> <p>Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research 15:1929–1958.</p> <p>Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. Training very deep networks. In NIPS.</p> <p>Joseph P. Turian, Lev-Arie Ratinov, and Yoshua Bengio. 2010. Word representations: A simple and general method for semi-supervised learning.</p>	
---	--

<p>In ACL.</p> <p>Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. 2017. Gated self-matching networks for reading comprehension and question answering. In ACL.</p> <p>John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2016. Charagram: Embedding words and sentences via character n-grams. In EMNLP.</p> <p>Sam Wiseman, Alexander M. Rush, and Stuart M. Shieber. 2016. Learning global features for coreference resolution. In HLT-NAACL.</p> <p>Matthew D. Zeiler. 2012. Adadelata: An adaptive learning rate method. CoRR abs/1212.5701.</p> <p>Jie Zhou and Wei Xu. 2015. End-to-end learning of semantic role labeling using recurrent neural networks. In ACL.</p> <p>Peng Zhou, Zhenyu Qi, Suncong Zheng, Jiaming Xu, Hongyun Bao, and Bo Xu. 2016. Text classification improved by integrating bidirectional lstm with twodimensional max pooling. In COLING.</p>	
--	--