DEEP LEARNING FOR NATURAL LANGUAGE PROCESSING

Sergey I. Nikolenko[1,2,3]

AI Ukraine
Kharkiv, Ukraine, October 8, 2016

[1] NRU Higher School of Economics, St. Petersburg

[2] Steklov Institute of Mathematics at St. Petersburg

[3] Deloitte Analytics Institute, Moscow

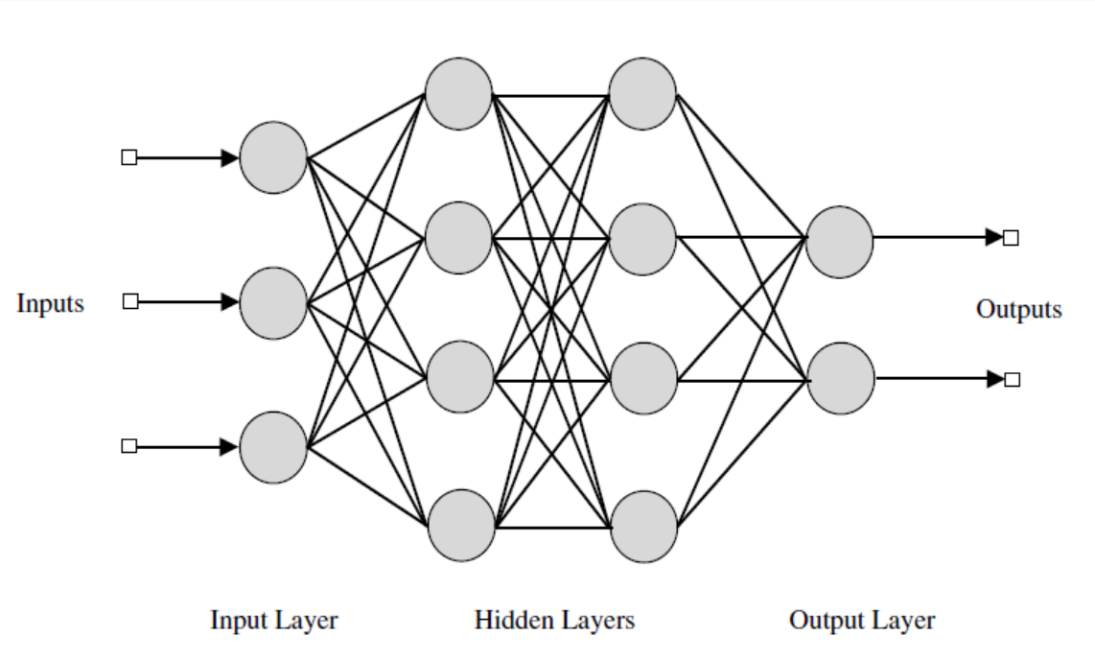
Random facts: on October 8, 1480, the Great Standoff on the Ugra River ended Tatar rule in Russia; on October 8, 1886, the first public library opened in Kharkiv (now named after Korolenko).

PLAN

- The deep learning revolution has not left natural language processing alone.
- DL in NLP has started with standard architectures (RNN, CNN) but then has branched out into new directions.
- Our plan for today:
  (1) a primer on sentence embeddings and character-level models;
  (2) a ((very-)very) brief overview of the most promising directions in modern NLP based on deep learning.
- We will concentrate on directions that have given rise to new models and architectures.

BASIC NN ARCHITECTURES

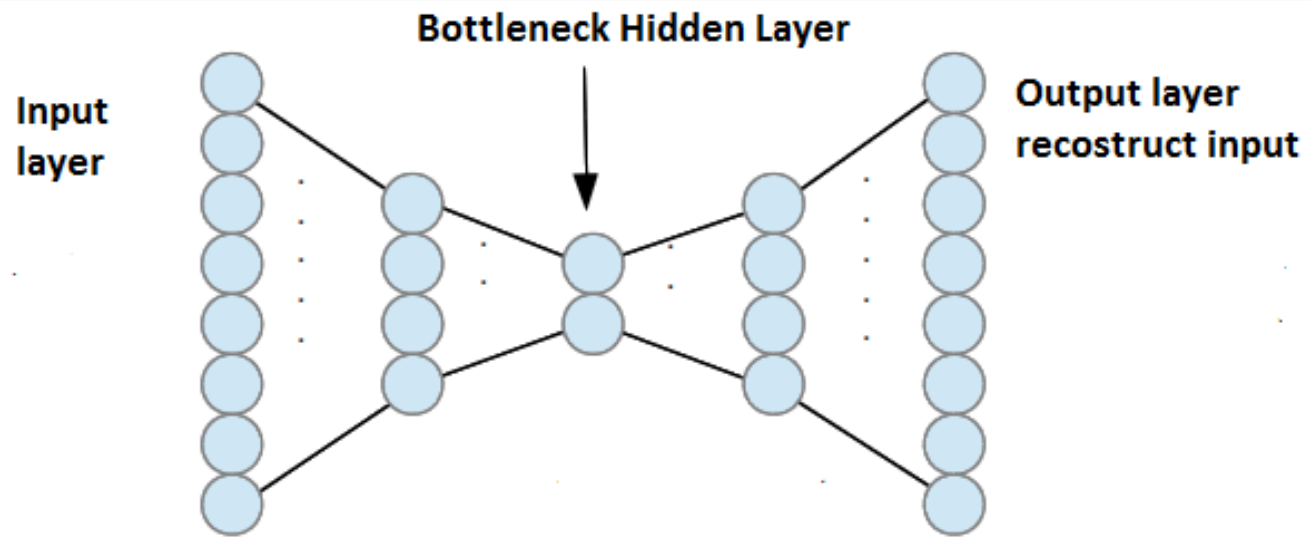
- Basic neural network architectures that have been adapted for deep learning over the last decade:
  - *feedforward* NNs are the basic building block;



- *Deep* learning refers to several layers, any network mentioned above can be deep or shallow, usually in several different ways.
- So let us see how all this comes into play for natural language...

BASIC NN ARCHITECTURES

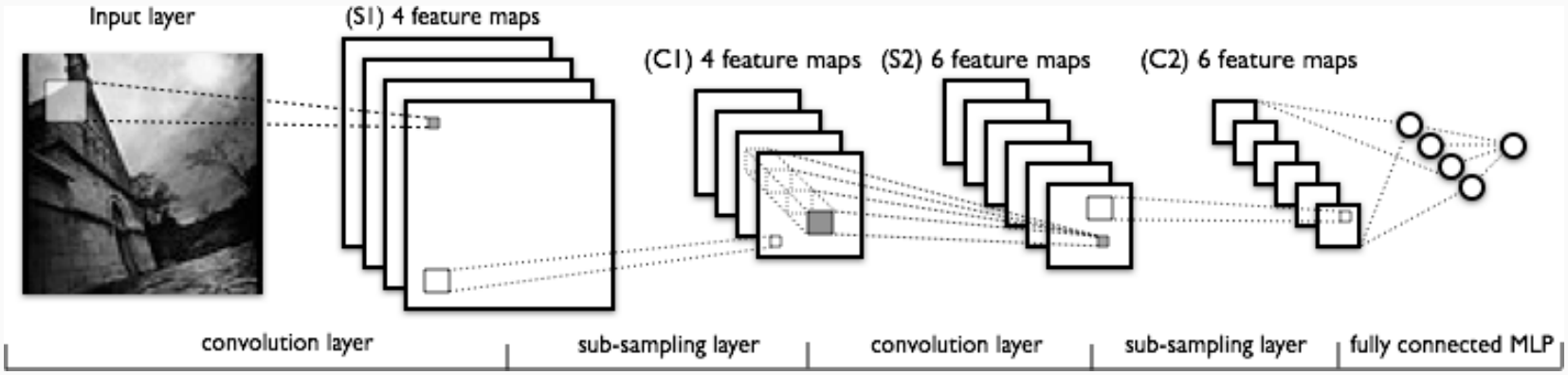
- Basic neural network architectures that have been adapted for deep learning over the last decade:
  - *autoencoders* map a (possibly distorted) input to itself, usually for feature engineering;



- *Deep* learning refers to several layers, any network mentioned above can be deep or shallow, usually in several different ways.
- So let us see how all this comes into play for natural language...

BASIC NN ARCHITECTURES

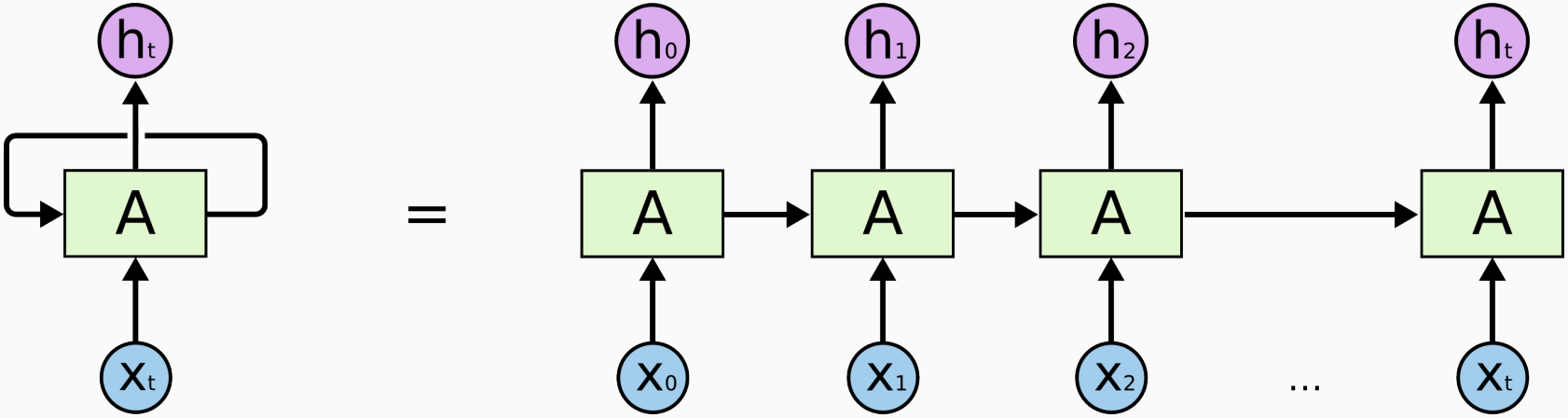
- Basic neural network architectures that have been adapted for deep learning over the last decade:
  - *convolutional* NNs apply NNs with shared weights to certain windows in the previous layer (or input), collecting first local and then more and more global features;



- *Deep* learning refers to several layers, any network mentioned above can be deep or shallow, usually in several different ways.
- So let us see how all this comes into play for natural language...

BASIC NN ARCHITECTURES

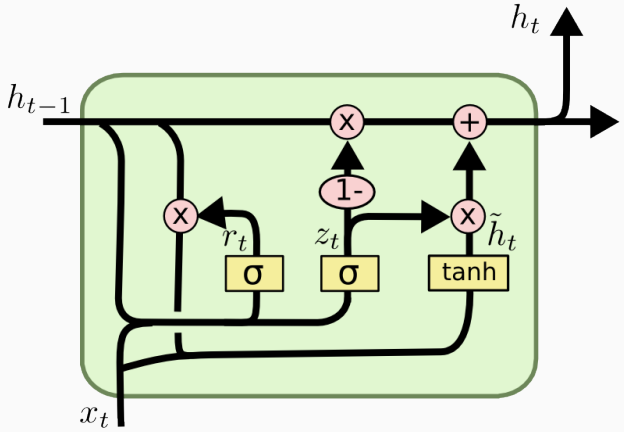
- Basic neural network architectures that have been adapted for deep learning over the last decade:
  - *recurrent* NNs have a hidden state and propagate it further, used for sequence learning;



- *Deep* learning refers to several layers, any network mentioned above can be deep or shallow, usually in several different ways.
- So let us see how all this comes into play for natural language...

BASIC NN ARCHITECTURES

- Basic neural network architectures that have been adapted for deep learning over the last decade:
  - in particular, LSTM (*long short-term memory*) and GRU (*gated recurrent unit*) units are an important RNN architecture often used for NLP, good for longer dependencies.



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

- *Deep* learning refers to several layers, any network mentioned above can be deep or shallow, usually in several different ways.
- So let us see how all this comes into play for natural language...

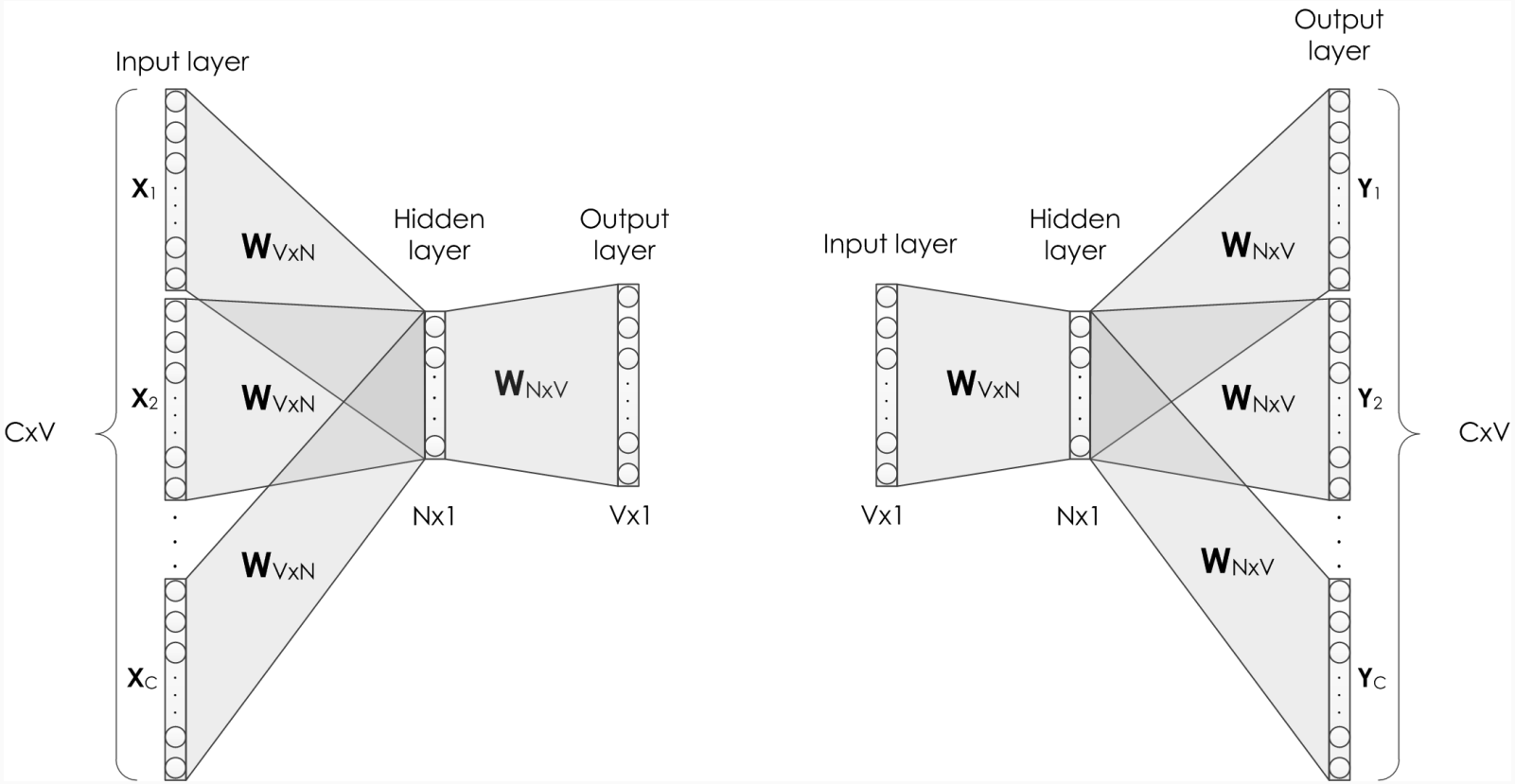
WORD EMBEDDINGS, SENTENCE EMBEDDINGS, AND CHARACTER-LEVEL MODELS

WORD EMBEDDINGS

- Distributional hypothesis in linguistics: words with similar meaning will occur in similar contexts.
- *Distributed word representations* map words to a Euclidean space (usually of dimension several hundred):
  - started in earnest in (Bengio et al. 2003; 2006), although there were earlier ideas;
  - *word2vec* (Mikolov et al. 2013): train weights that serve best for simple prediction tasks between a word and its context: continuous bag-of-words (CBOW) and skip-gram;
  - *Glove* (Pennington et al. 2014): train word weights to decompose the (log) cooccurrence matrix.
- Interestingly, semantic relationships between the words sometimes map into geometric relationships:
  $\text{king} + \text{woman} - \text{man} \approx \text{queen}$,
  $\text{Moscow} + \text{France} - \text{Russia} \approx \text{Paris}$, and so on.

CBOW AND SKIP-GRAM

- Difference between skip-gram and CBOW architectures:
  - CBOW model predicts a word from its local context;
  - skip-gram model predicts context words from the current word.



WORD EMBEDDING EXAMPLES

- Russian examples:
  - nearest neighbors of the word **конференция**:

пресс-конференция	0.6919
программа	0.6827
выставка	0.6692
ассоциация	0.6638
кампания	0.6406
ярмарка	0.6372
экспедиция	0.6305
презентация	0.6243
сходка	0.6162
встреча	0.6100

WORD EMBEDDING EXAMPLES

- Sometimes antonyms also fit:
  - nearest neighbors of the word **любовь**:

жизнь	0.5978
нелюбовь	0.5957
приязнь	0.5735
боль	0.5547
страсть	0.5520

  - nearest neighbors of the word **синоним**:

антоним	0.5459
эвфемизм	0.4642
анаграмма	0.4145
омоним	0.4048
оксюморон	0.3930

WORD EMBEDDING EXAMPLES

- On sexism:
  - nearest neighbors of the word **программист**:

компьютерщик	0.5618
программер	0.4682
электронщик	0.4613
автомеханик	0.4441
криптограф	0.4316

  - nearest neighbors of the word **программистка**:

стажерка	0.4755
инопланетянка	0.4500
американочка	0.4481
предпринимательница	0.4442
студенточка	0.4368

WORD EMBEDDING EXAMPLES

- What do you think are the
  - nearest neighbors of the word **комендантский**?

WORD EMBEDDING EXAMPLES

- What do you think are the
  - nearest neighbors of the word **комендантский**:

неурочный	0.7276
неровен	0.7076
урочный	0.6849
ровен	0.6756
предрассветный	0.5867
условленный	0.5597

UP AND DOWN FROM WORD EMBEDDINGS

- Word embeddings are the first step of most DL models in NLP.
- But we can go both up and down from word embeddings.
- First, a sentence is not necessarily the sum of its words.
- Second, a word is not quite as atomic as the word2vec model would like to think.

SENTENCE EMBEDDINGS

- How do we combine word vectors into “text chunk” vectors?
- The simplest idea is to use the sum and/or mean of word embeddings to represent a sentence/paragraph:
  - a baseline in (Le and Mikolov 2014);
  - a reasonable method for short phrases in (Mikolov et al. 2013)
  - shown to be effective for document summarization in (Kageback et al. 2014).

SENTENCE EMBEDDINGS

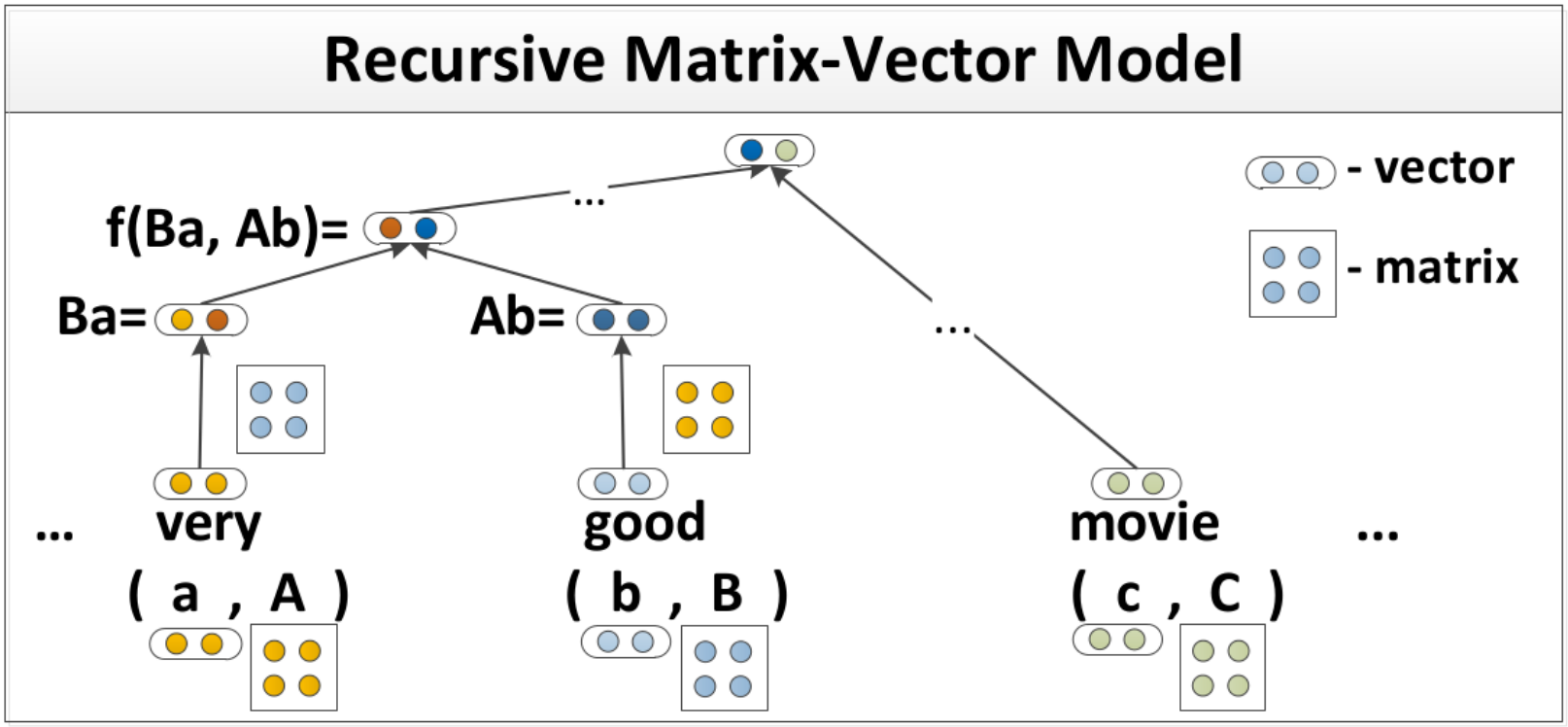
- How do we combine word vectors into “text chunk” vectors?
- Distributed Memory Model of Paragraph Vectors (PV-DM) (Le and Mikolov 2014):
  - a sentence/paragraph vector is an additional vector for each paragraph;
  - acts as a “memory” to provide longer context;
- Distributed Bag of Words Model of Paragraph Vectors (PV-DBOW) (Le and Mikolov 2014):
  - the model is forced to predict words randomly sampled from a specific paragraph;
  - the paragraph vector is trained to help predict words from the same paragraph in a small window.

SENTENCE EMBEDDINGS

- How do we combine word vectors into “text chunk” vectors?
- A number of convolutional architectures (Ma et al., 2015; Kalchbrenner et al., 2014).
- (Kiros et al. 2015): skip-thought vectors capture the meanings of a sentence by training from skip-grams constructed on sentences.
- (Djuric et al. 2015): model large text streams with hierarchical neural language models with a document level and a token level.

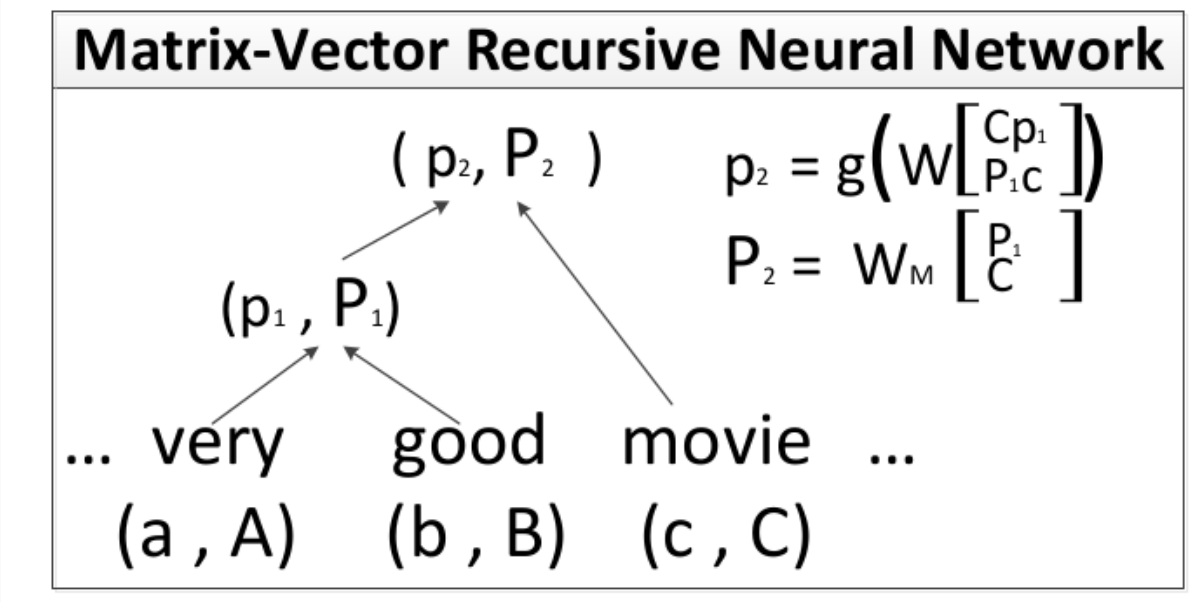
SENTENCE EMBEDDINGS

- How do we combine word vectors into “text chunk” vectors?
- *Recursive neural networks* (Socher et al., 2012):
  - a neural network composes a chunk of text with another part in a tree;
  - works its way up from word vectors to the root of a parse tree.



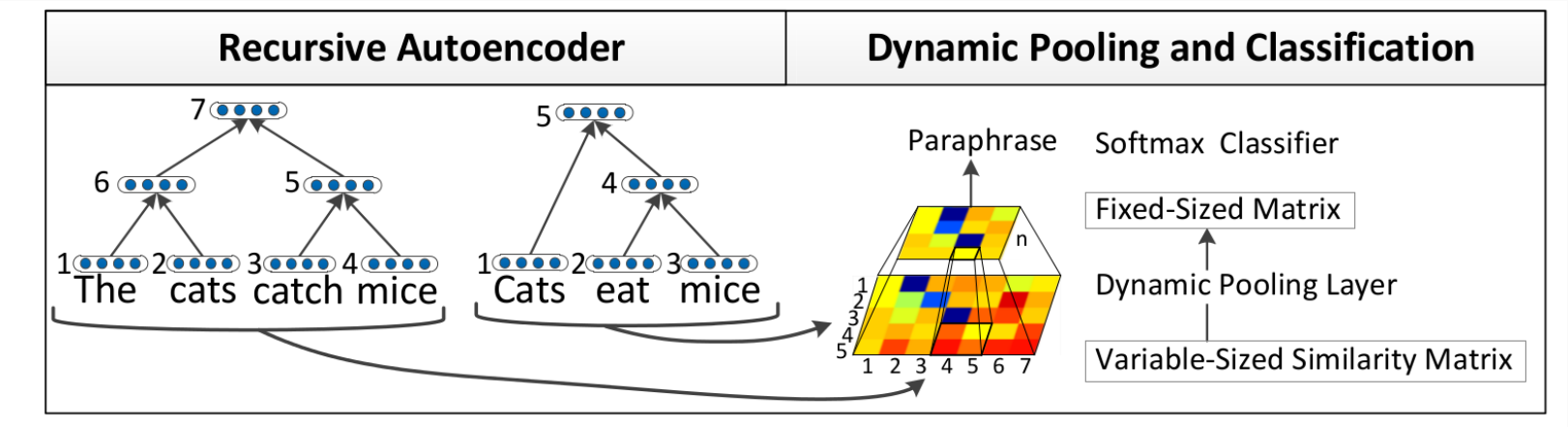
SENTENCE EMBEDDINGS

- How do we combine word vectors into “text chunk” vectors?
- *Recursive neural networks* (Socher et al., 2012):
  - by training this in a supervised way, one can get a very effective approach to sentiment analysis (Socher et al. 2013).



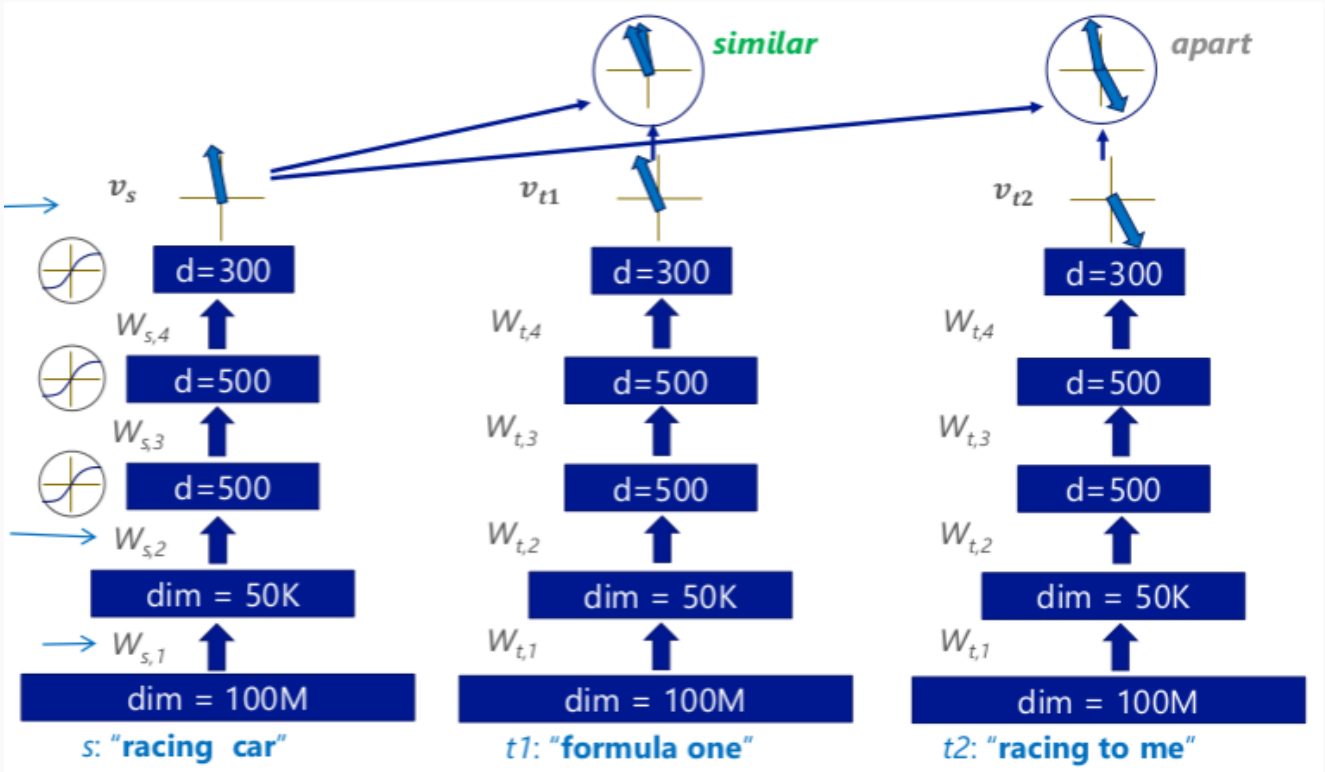
SENTENCE EMBEDDINGS

- How do we combine word vectors into “text chunk” vectors?
- A similar effect can be achieved with CNNs.
- *Unfolding Recursive Auto-Encoder* model (URAE) (Socher et al., 2011) collapses all word embeddings into a single vector following the parse tree and then reconstructs back the original sentence; applied to paraphrasing and paraphrase detection.



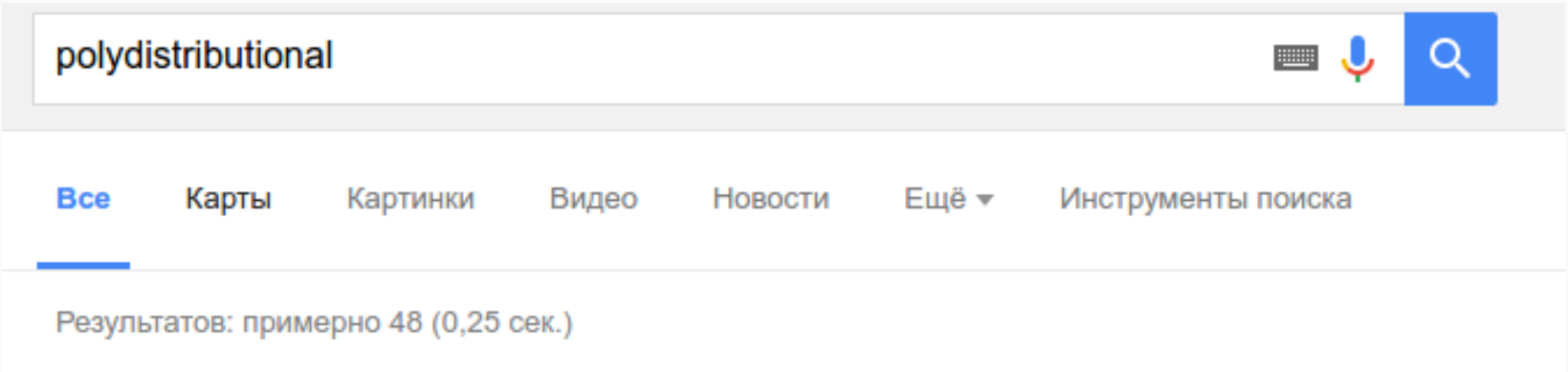
SENTENCE EMBEDDINGS

- How do we combine word vectors into “text chunk” vectors?
- *Deep Structured Semantic Models* (DSSM) (Huang et al., 2013; Gao et al., 2014a; 2014b): a deep convolutional architecture trained on similar text pairs.



CHARACTER-LEVEL MODELS

- Word embeddings have important shortcomings:
  - vectors are independent but words are not; consider, in particular, morphology-rich languages like Russian/Ukrainian;
  - the same applies to out-of-vocabulary words: a word embedding cannot be extended to new words;
  - word embedding models may grow large; it's just lookup, but the whole vocabulary has to be stored in memory with fast access.
- E.g., "polydistributional" gets 48 results on Google, so you probably have never seen it, and there's very little training data:



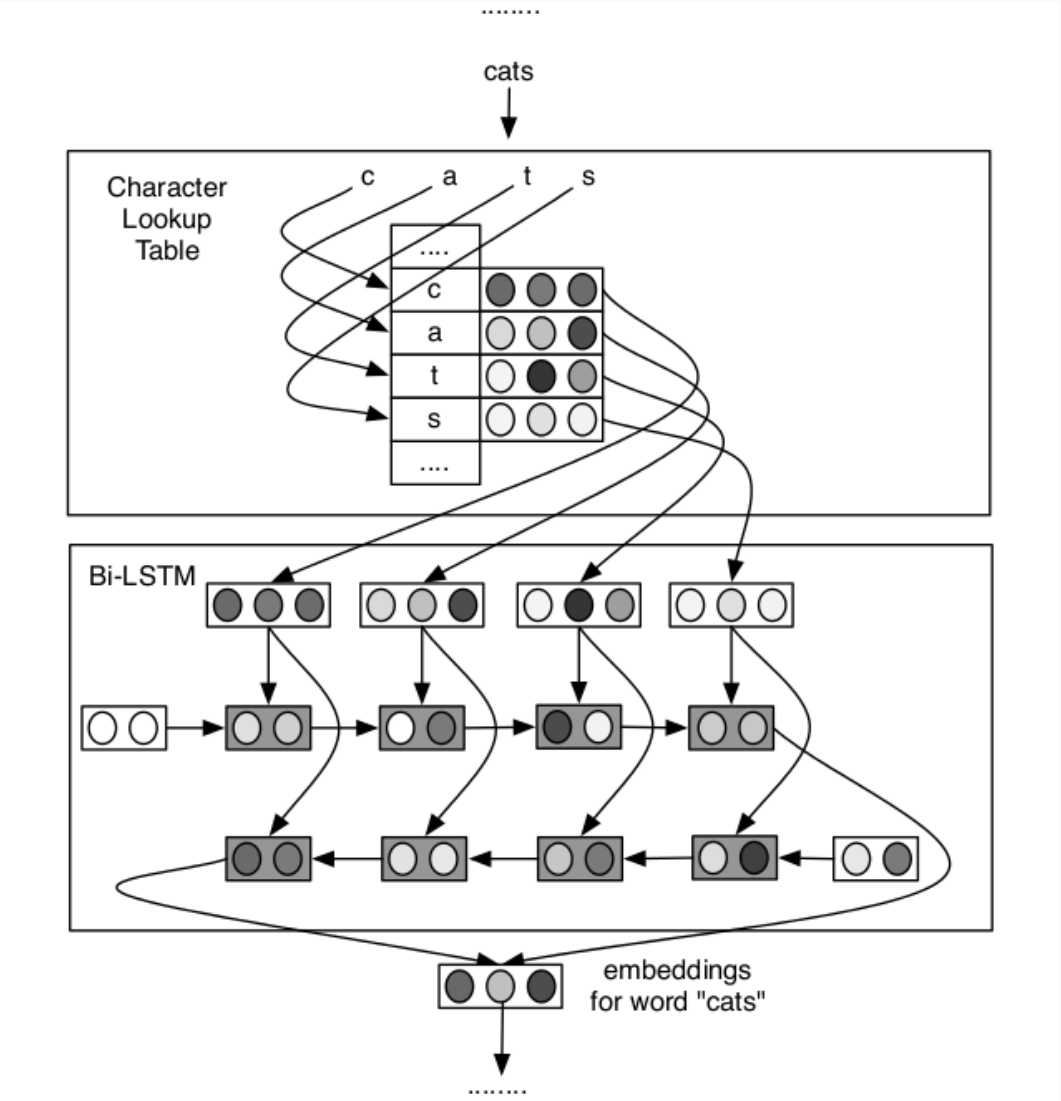
- Do you have an idea what it means? Me too.

Character-level models

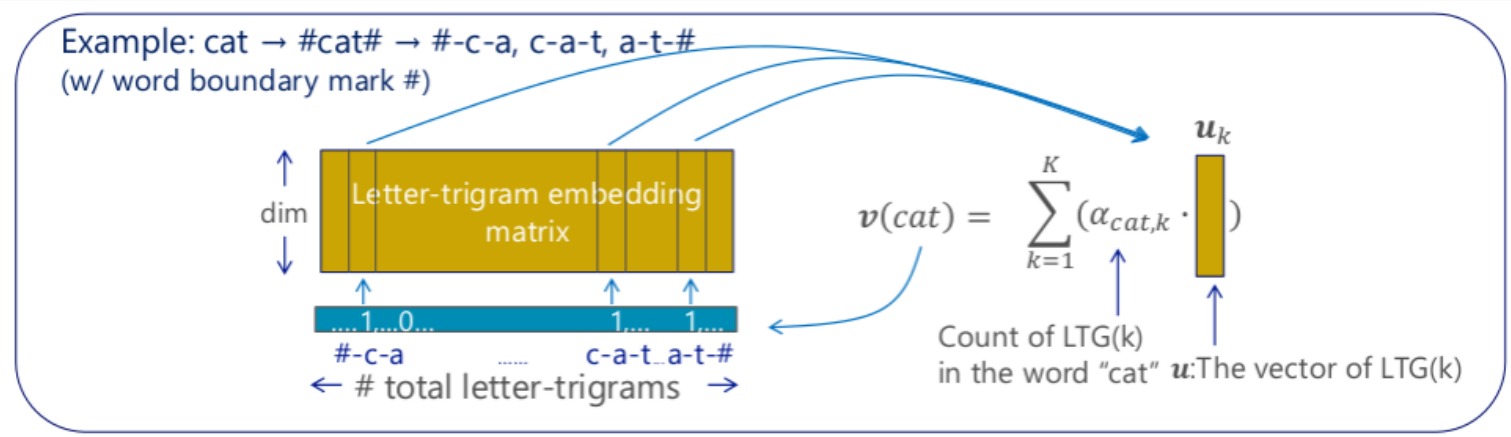
- Hence, *character-level representations*:
  - began by decomposing a word into morphemes (Luong et al. 2013; Botha and Blunsom 2014; Soricut and Och 2015);
  - but this adds errors since morphological analyzers are also imperfect, and basically a part of the problem simply shifts to training a morphology model;
  - two natural approaches on character level: LSTMs and CNNs;
  - in any case, the model is slow but we do not have to apply it to every word, we can store embeddings of common words in a lookup table as before and only run the model for rare words – a nice natural tradeoff.

CHARACTER-LEVEL MODELS

- C2W (Ling et al. 2015) is based on bidirectional LSTMs:

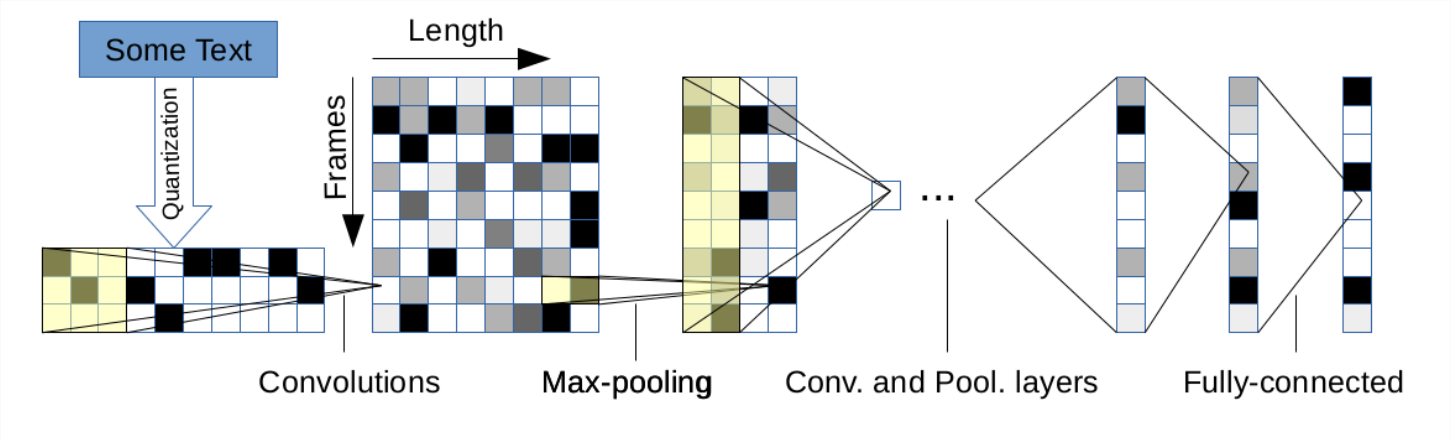


- The approach of *Deep Structured Semantic Model* (DSSM) (Huang et al., 2013; Gao et al., 2014a; 2014b):
  - sub-word embeddings: represent a word as a bag of trigrams;
  - vocabulary shrinks to $|V|^3$ (tens of thousands instead of millions), but collisions are very rare;
  - the representation is robust to misspellings (very important for user-generated texts).



CHARACTER-LEVEL MODELS

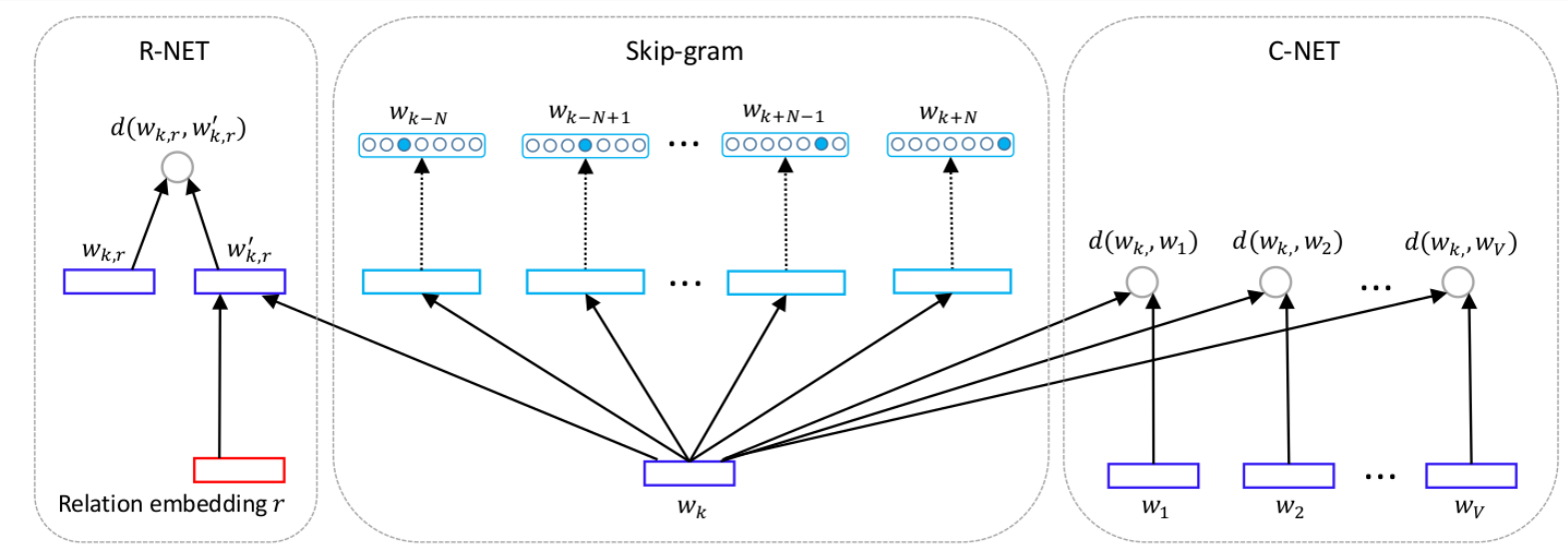
- ConvNet (Zhang et al. 2015): text understanding from scratch, from the level of symbols, based on CNNs.
- They also propose a nice idea for data augmentation (replace words with synonyms from WordNet or such).



- Character-level models and extensions to appear to be very important, especially for morphology-rich languages like Russian/Ukrainian.

CHARACTER-LEVEL MODELS

- Other modifications of word embeddings add external information.
- E.g., the RC-NET model (Xu et al. 2014) extends skip-grams with relations (semantic and syntactic) and categorical knowledge (sets of synonyms, domain knowledge etc.).



WORD SENSE DIAMBIGUATION

- Another important problem with both word vectors and char-level models: homonyms.
- How do we distinguish different senses of the same word?
  - most similar for the word **коса**:

бахрома	0.4898
косичка	0.4867
пластина	0.4738
балка	0.4616
проплешина	0.4585

  - but really usually the model just chooses one meaning.
- We have to add *latent* variables for different meaning and infer them from context.
- To train the meanings with latent variables — Bayesian inference with stochastic variational inference (Bartunov et al., 2015).

GENERAL APPROACHES

TEXT GENERATION WITH RNNS

- Language modeling and text generation is a natural direct application of NN-based NLP; word embeddings started as a "neural probabilistic language model" (Bengio et al., 2003).
- First idea – sequence learning with RNNs/LSTMs.
- Surprisingly, simple RNNs can produce quite reasonably-looking text even by training character by character, with no knowledge of the words ("The Unreasonable Effectiveness..."), including the famous example from (Sutskever et al. 2011):

  The meaning of life is the tradition of the ancient human reproduction: it is less favorable to the good boy for when to remove her bigger...
- This is, of course, not "true understanding" (whatever that means), only short-term memory effects.
- We need to go deeper in terms of both representations and sequence modeling.

TEXT GENERATION WITH RNNS

- One can change diversity (sampling temperature) and get different styles of absurdistic texts. Random example with seed «*обещал, на рождество, но спустя семь лет. имя глав*».
- Low diversity gets you Lucky's monologue from "Waiting for Godot":

обещал, на рождество, но спустя семь лет. имя главного командования в составе советского союза с 1976 года. после проведения в 1992 году в составе советского союза (1977). в 1967 году в составе советского союза состоялся в 1952 году в составе советско-финской войны 1877 - 1877 годов. в 1966 году в составе советского союза с 1965 года по 1977 год...

- Mid-range diversity produces something close to meaningful text:

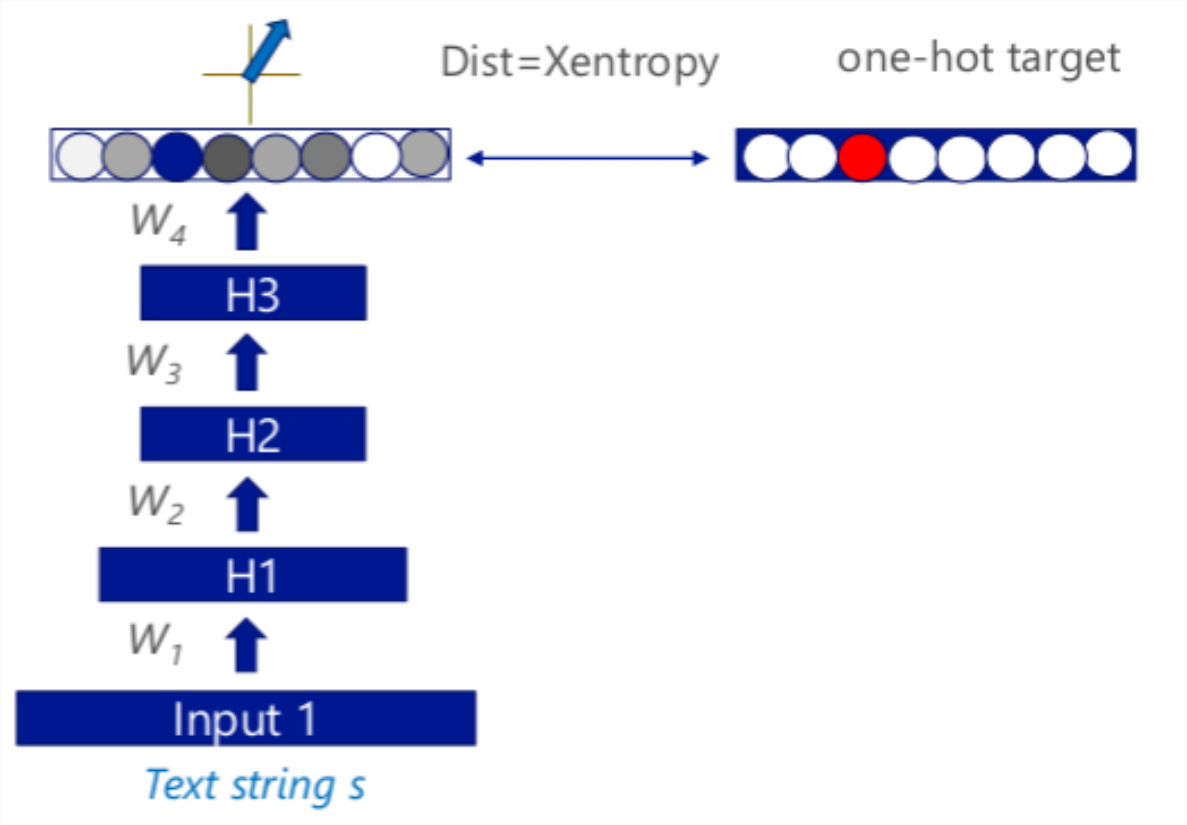
обещал, на рождество, но спустя семь лет. имя главного рода собственно вновь образовалась в россии и народном состоянии. после присказа с постановлением союза писателей россии и генеральной диссертации о спортивном училище с 1980 года. в 1970-х годах был основан в составе комитета высшего совета театра в польши. в 1957 - 1962 годах - начальник батальона сан-аухаров...

- High diversity leads to Khlebnikov's zaum:

обещал, на рождество, но спустя семь лет. имя главы философии пововпели nol-lнози - врайу-7 на луосече. человеческая восстания покторов извоенного чомпде и э. дроссенбурга, ... карл уним-общекрипских. эйелем хфечак от этого списка сравнивала имущно моря в юнасториансический индристское носительских женатов в церкви испании....

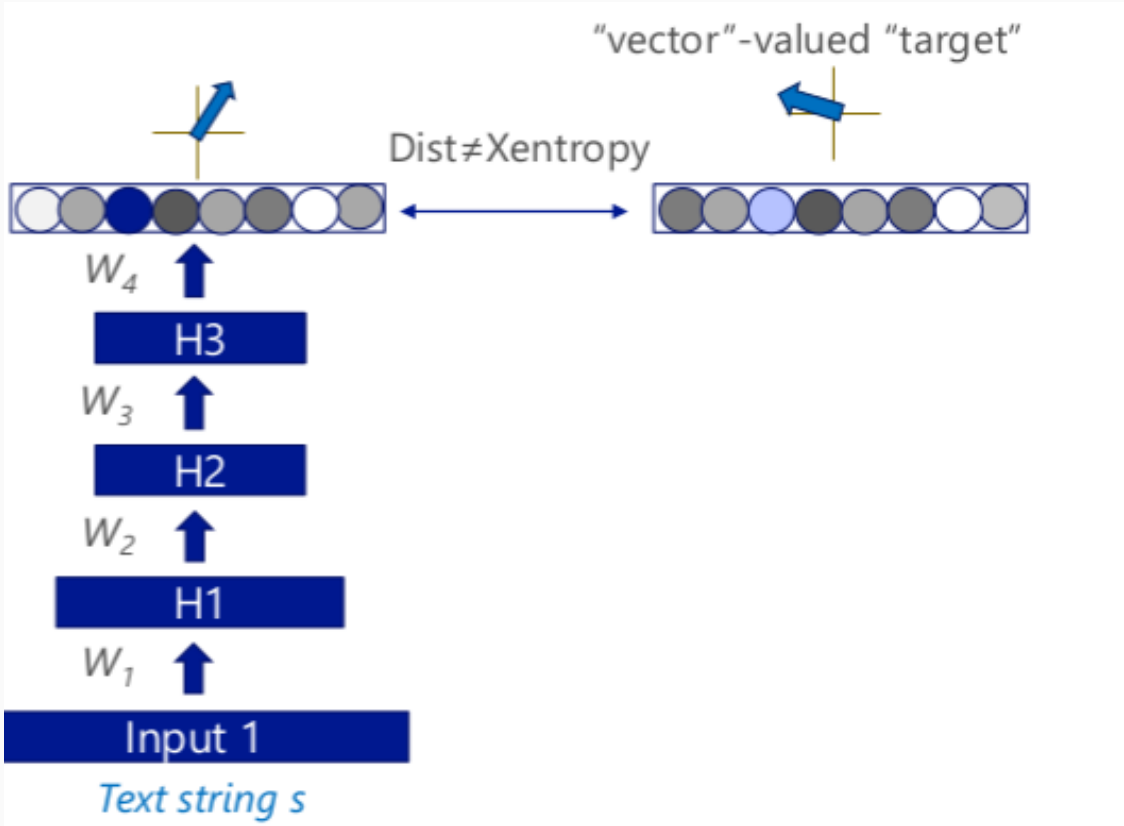
DSSM

- A general approach to NLP based on CNNs is given by *Deep Structured Semantic Model* (DSSM) (Huang et al., 2013; Gao et al., 2014a; 2014b):
  - one-hot target vectors for classification (speech recognition, image recognition, language modeling).



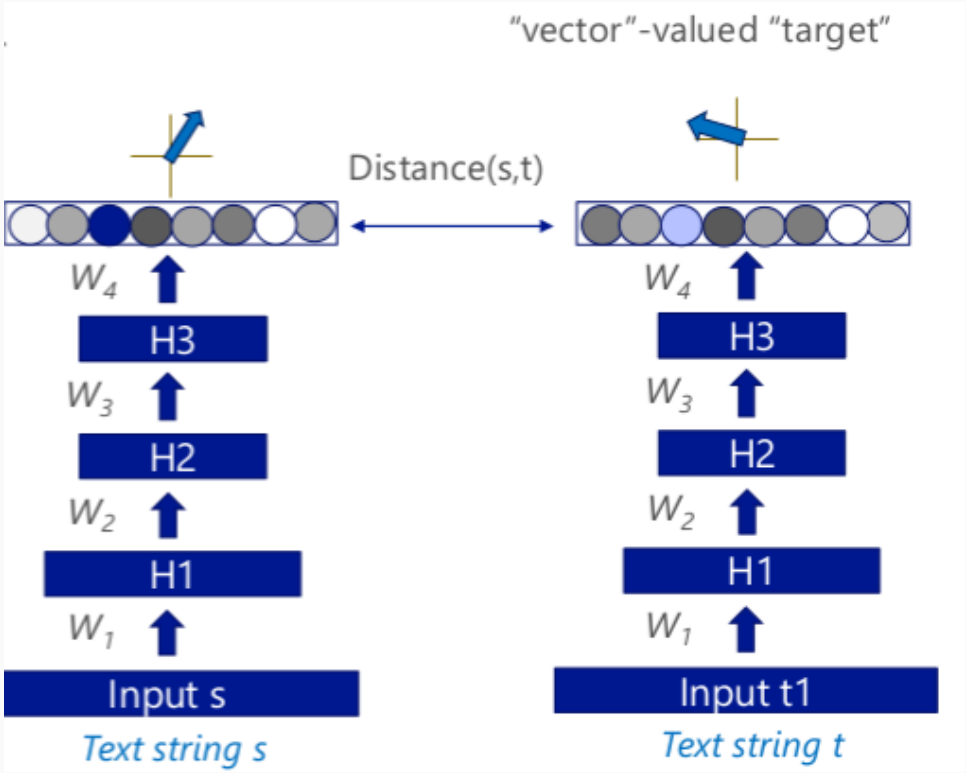
DSSM

- A general approach to NLP based on CNNs is given by *Deep Structured Semantic Model* (DSSM) (Huang et al., 2013; Gao et al., 2014a; 2014b):
  - vector-valued targets for semantic matching.



DSSM

- A general approach to NLP based on CNNs is given by *Deep Structured Semantic Model* (DSSM) (Huang et al., 2013; Gao et al., 2014a; 2014b):
  - can capture different targets (one-hot, vector);
  - to train with vector targets – reflection: bring source and target vectors closer.

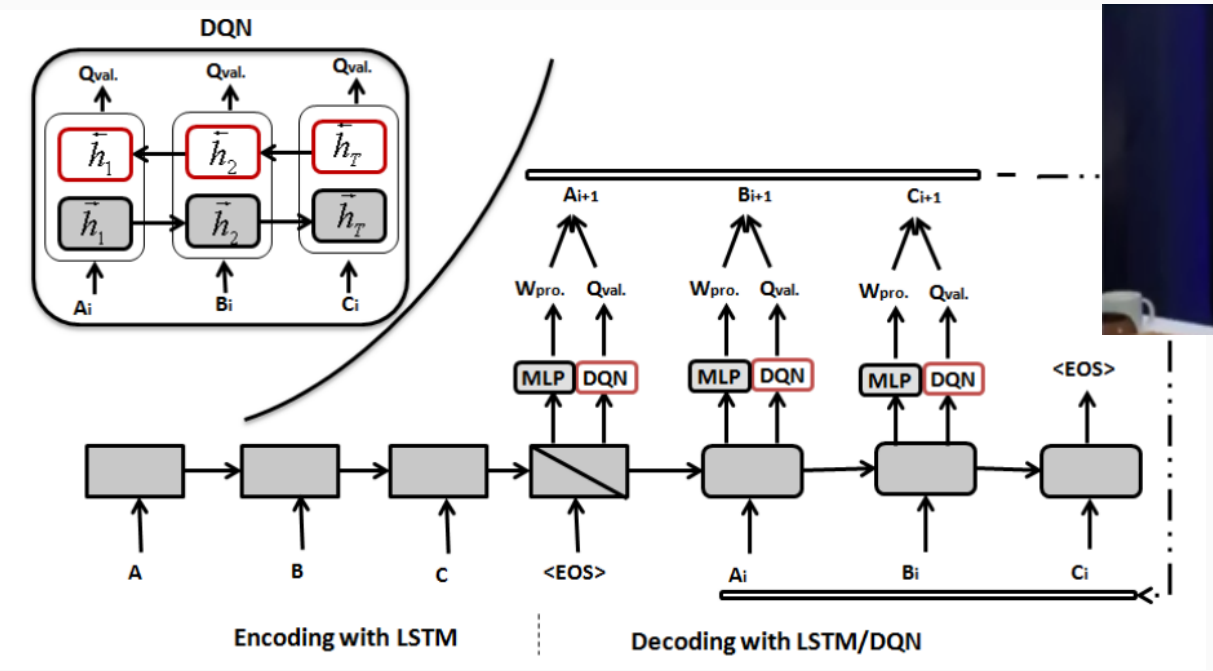


DSSM

- A general approach to NLP based on CNNs is given by *Deep Structured Semantic Model* (DSSM) (Huang et al., 2013; Gao et al., 2014a; 2014b):
- DSSMs can be applied in a number of different contexts when we can specify a supervised dataset:
  - semantic word embeddings: word by context;
  - web search: web documents by query;
  - question answering: knowledge base relation/entity by pattern;
  - recommendations: interesting documents by read/liked documents;
  - translation: target sentence by source sentence;
  - text/image: labels by images or vice versa.
- Basically, this is an example of a general architecture that can be trained to do almost anything.

DQN

- (Guo, 2015): generating text with deep reinforcement learning.
- Begin with easy parts, then iteratively decode the hard parts with DQN.



- Next, we proceed to specific NLP problems that have led to interesting developments.

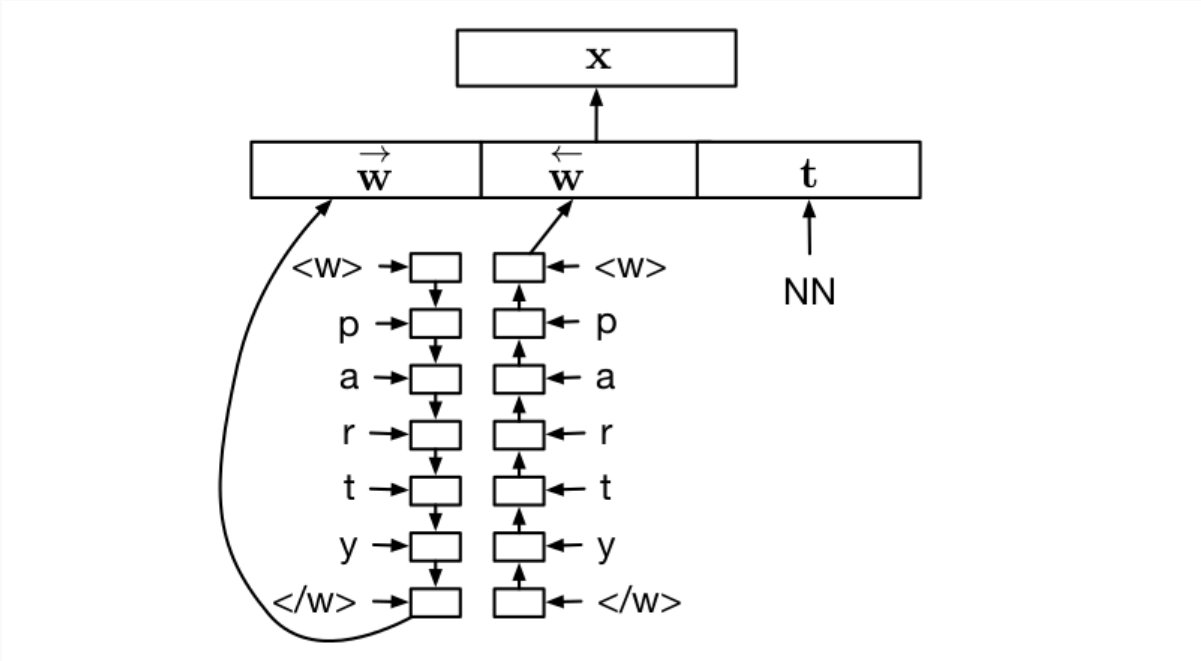
DEPENDENCY PARSING

DEPENDENCY PARSING

- We mentioned parse trees; but how do we construct them?
- Current state of the art – continuous-state parsing: current state is encoded in $\mathbf{R}^{\mathbf{d}}$.
- *Stack LSTMs* (Dyer et al., 2015) – the parser manipulates three basic data structures:
  (1) a buffer $\mathbf{B}$ that contains the sequence of words, with state $\mathbf{b_t}$;
  (2) a stack $\mathbf{S}$ that stores partially constructed parses, with state $\mathbf{s_t}$;
  (3) a list $\mathbf{A}$ of actions already taken by the parser, with state $\mathbf{a_t}$.
- $\mathbf{b_t}$, $\mathbf{s_t}$, and $\mathbf{a_t}$ are hidden states of stack LSTMs, LSTMs that have a stack pointer: new inputs are added from the right, but the current location of the stack pointer shows which cell's state is used to compute new memory cell contents.

DEPENDENCY PARSING WITH MORPHOLOGY

- Important extension – (Ballesteros et al., 2015):
  - in morphologically rich natural languages, we have to take into account morphology;
  - so they represent the words by bidirectional character-level LSTMs;
  - report improved results in Arabic, Basque, French, German, Hebrew, Hungarian, Korean, Polish, Swedish, and Turkish;
  - this direction probably can be further improved (and where's Russian or Ukrainian in the list above?..).



Evaluation for sequence-to-sequence models

- Next we will consider specific models for machine translation, dialog models, and question answering.
- But how do we evaluate NLP models that produce text?
- Quality metrics for comparing with reference sentences produced by humans:
  - BLEU (Bilingual Evaluation Understudy): reweighted precision (incl. multiple reference translations);
  - METEOR: harmonic mean of unigram precision and unigram recall;
  - TER (Translation Edit Rate): number of edits between the output and reference divided by the average number of reference words;
  - LEPOR: combine basic factors and language metrics with tunable parameters.
- The same metrics apply to paraphrasing and, generally, all problems where the (supervised) answer should be a free-form text.

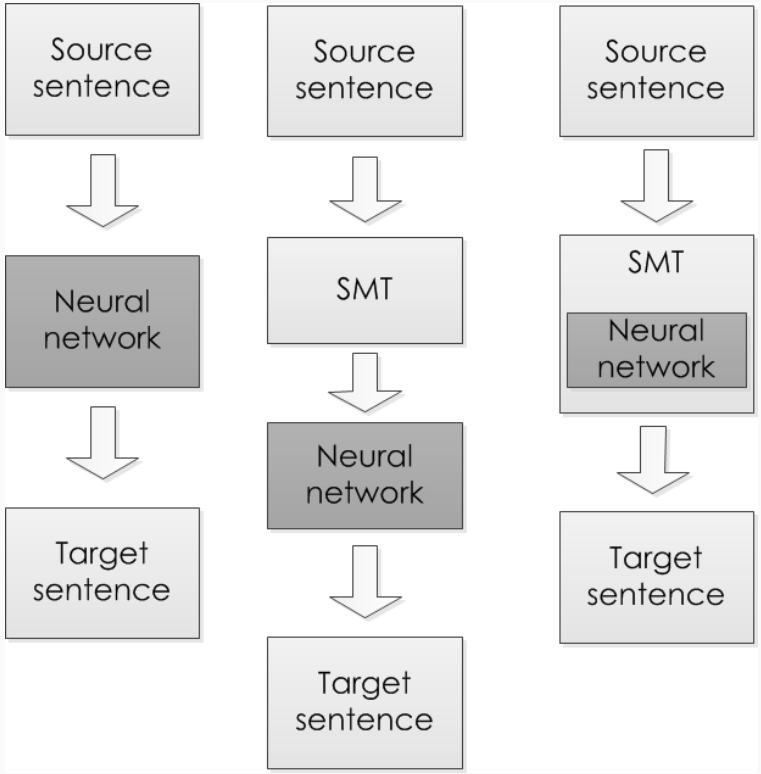
MACHINE TRANSLATION

MACHINE TRANSLATION

- Translation is a very convenient problem for modern NLP:
  - on one hand, it is very practical, obviously important;
  - on the other hand, it's very high-level, virtually impossible without deep understanding, so if we do well on translation, we probably do something right about understanding;
  - on the third hand (oops), it's quantifiable (BLEU, TER etc.) and has relatively large available datasets (parallel corpora).

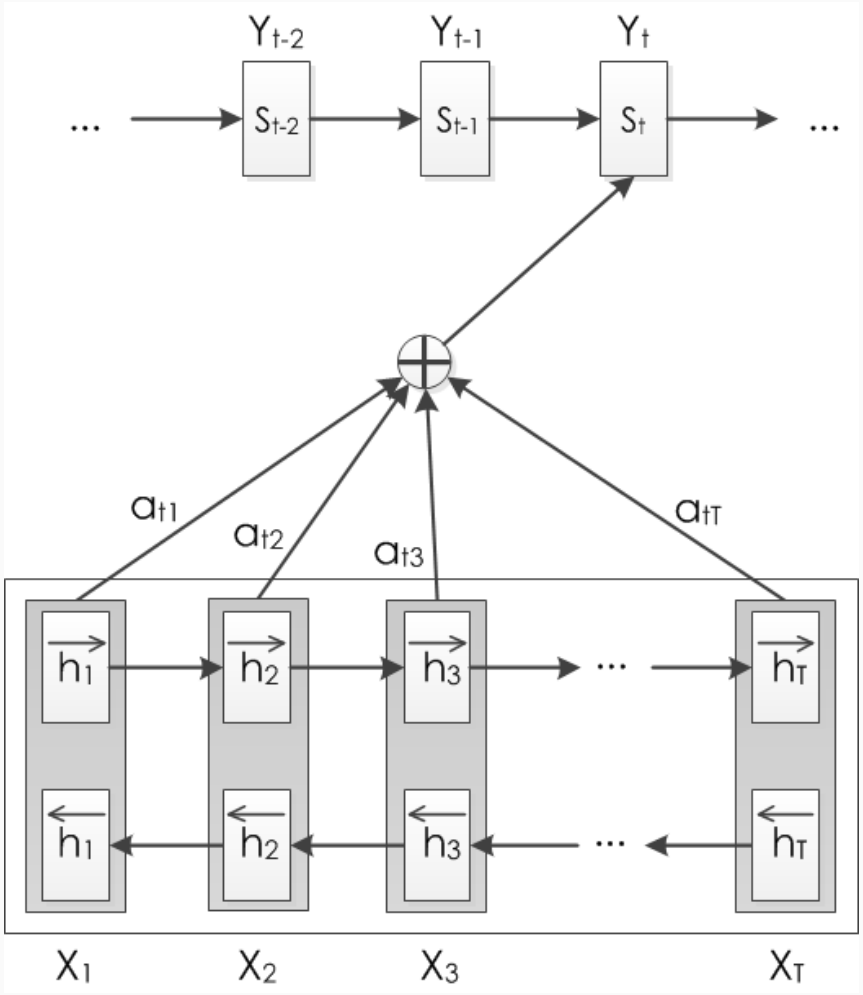
MACHINE TRANSLATION

- Statistical machine translation (SMT): model conditional probability $p(y \mid x)$ of target y (translation) given source x (text).
- Classical SMT: model $\log p(y \mid x)$ with a linear combination of features and then construct these features.
- NNs have been used both for reranking the best lists of possible translations and as part of feature functions:



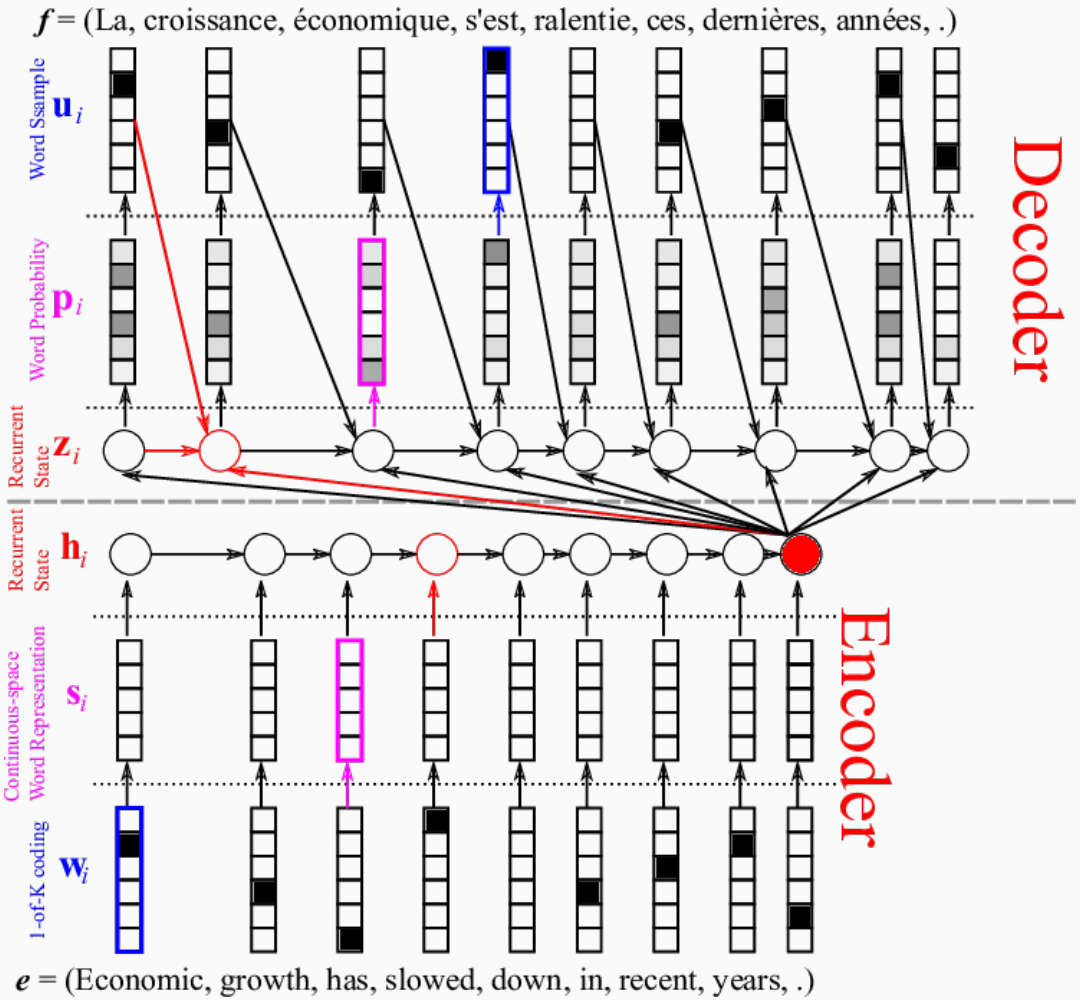
- NNs are still used for feature engineering with state of the art results, but here we are more interested in sequence-to-sequence modeling.
- Basic idea:
  - RNNs can be naturally used to probabilistically model a sequence $X = (x_1, x_2, \dots, x_T)$ as $p(x_1)$, $p(x_2 \mid x_1)$, ..., $p(x_T \mid x_{<T}) = p(x_T \mid x_{T-1}, \dots, x_1)$, and then the joint probability $p(X)$ is just their product $p(X) = p(x_1)p(x_2 \mid x_1) \dots p(x_k \mid x_{<k}) \dots p(x_T \mid x_{<T})$;
  - this is how RNNs are used for language modeling;
  - we predict next word based on the hidden state learned from all previous parts of the sequence;
  - in translation, maybe we can learn the hidden state from one sentence and apply to another.

- Direct application – bidirectional LSTMs (Bahdanau et al. 2014):

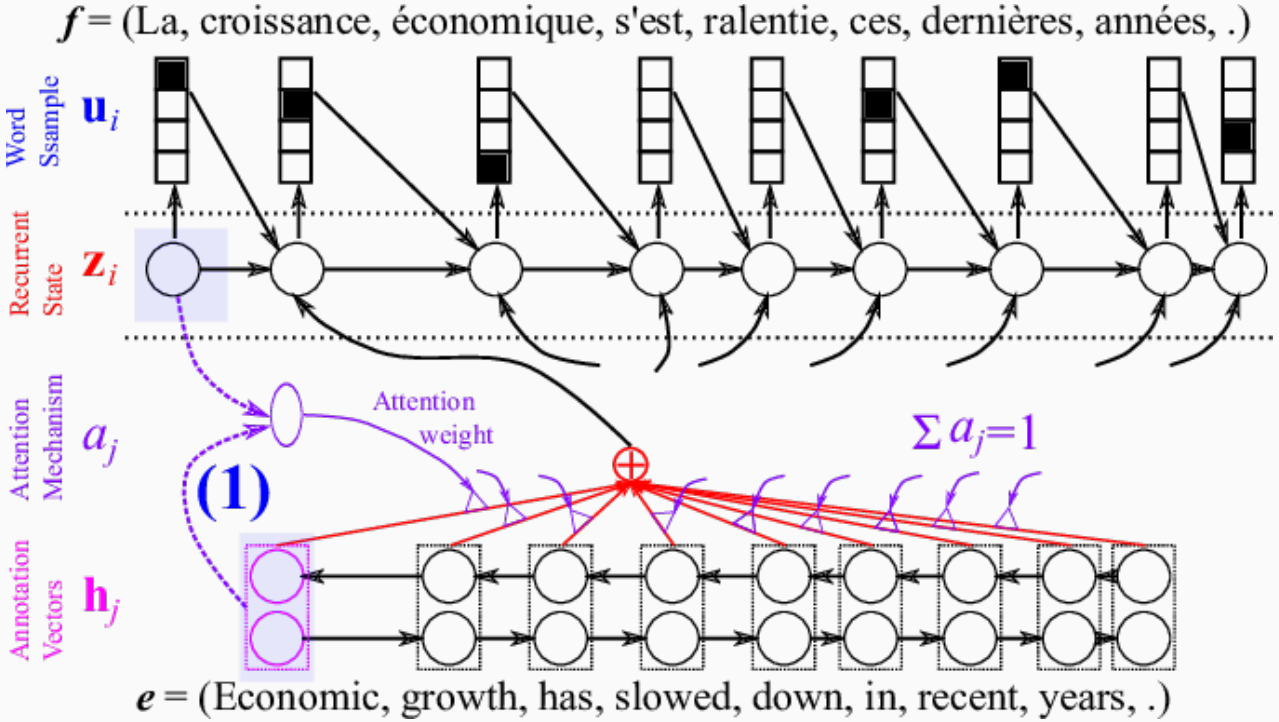


- But do we really translate word by word?

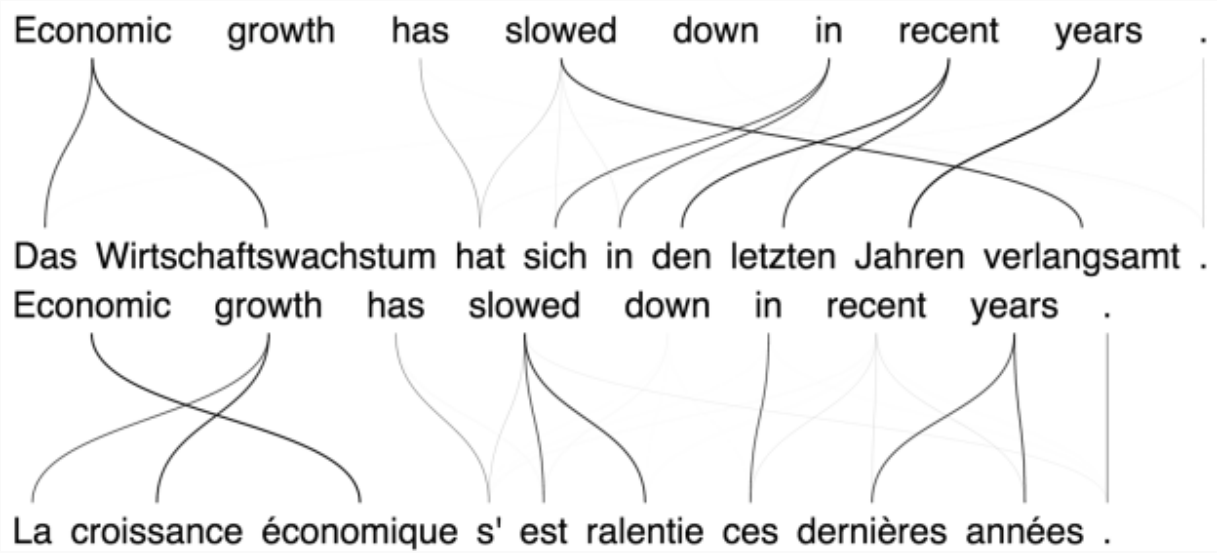
- No, we first understand the whole sentence; hence *encoder-decoder* architectures (Cho et al. 2014):



- But compressing the entire sentence to a fixed-dimensional vector is hard; quality drops dramatically with length.
- *Soft attention* (Luong et al. 2015a; 2015b; Jean et al. 2015):
  - encoder RNNs are bidirectional, so at every word we have a "focused" representation with both contexts;
  - attention NN takes the state and local representation and outputs a relevance score – should we translate this word right now?



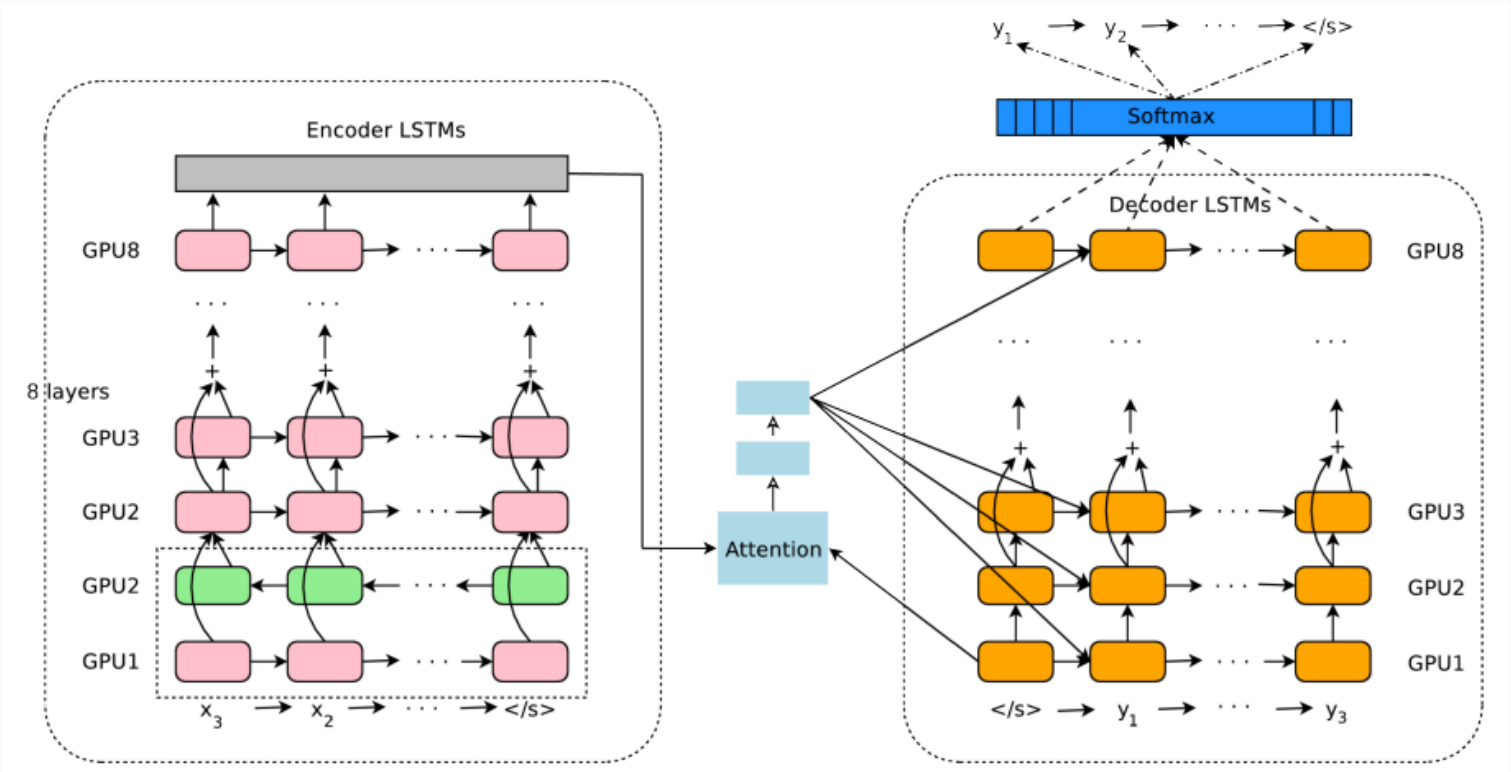
- We get better word order in the sentence as a whole:



- Attention is an “old” idea (Larochelle, Hinton, 2010), and can be applied to other RNN architectures, e.g., image processing and speech recognition; in other sequence-based NLP tasks:
  - syntactic parsing (Vinyals et al. 2014),
  - modeling pairs of sentences (Yin et al. 2015),
  - question answering (Hermann et al. 2015),
  - “Show, Attend, and Tell” (Xu et al. 2015).

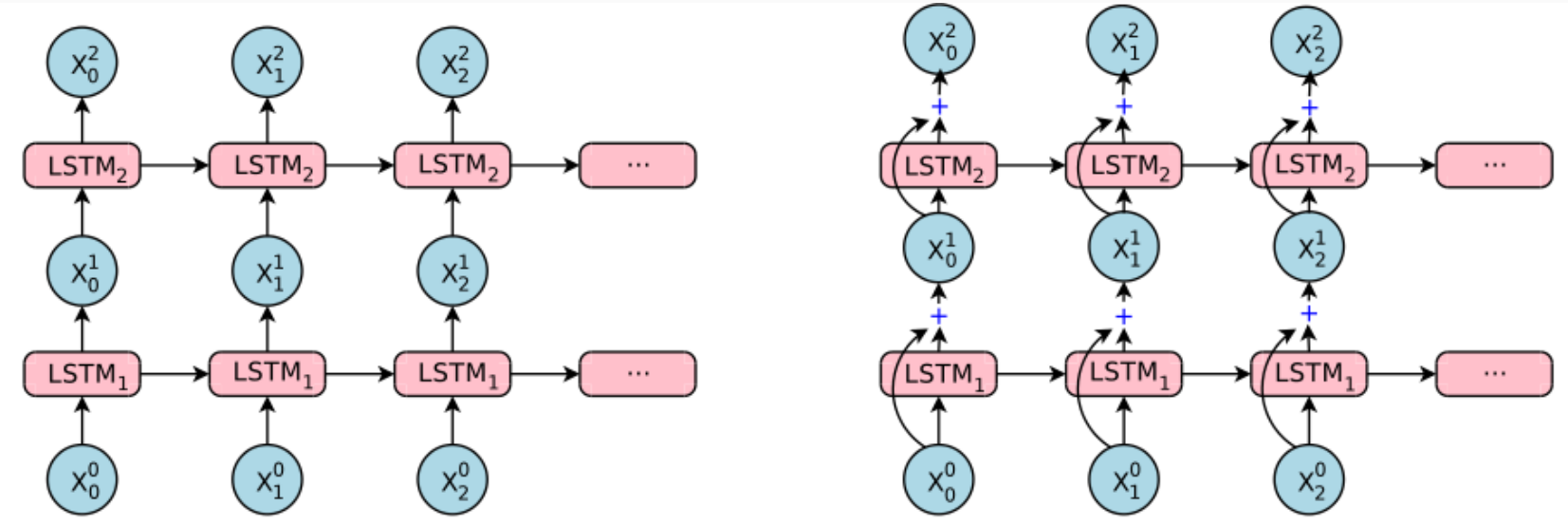
GOOGLE TRANSLATE

- Sep 26, 2016: Wu et al., *Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation*:
  - this very recent paper shows how Google Translate actually works;
  - the basic architecture is the same: encoder, decoder, attention;
  - RNNs have to be deep enough to capture language irregularities, so 8 layers for encoder and decoder each:



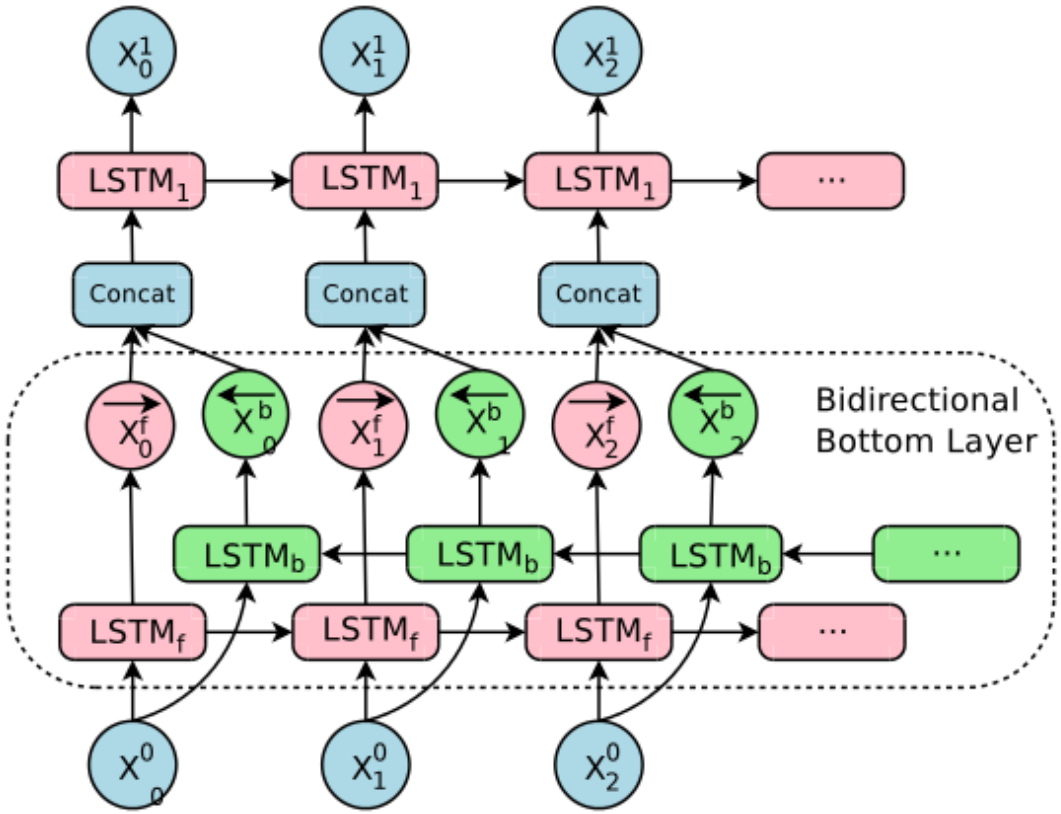
GOOGLE TRANSLATE

- Sep 26, 2016: Wu et al., *Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation*:
  - but stacking LSTMs does not really work: 4-5 layers are OK, 8 layers don't work;
  - so they add residual connections between the layers, similar to (He, 2015):



GOOGLE TRANSLATE

- Sep 26, 2016: Wu et al., *Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation*:
  - and it makes sense to make the bottom layer bidirectional in order to capture as much context as possible:



GOOGLE TRANSLATE

- Sep 26, 2016: Wu et al., *Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation*:
- GNMT also uses two ideas for word segmentation:
  - *wordpiece model*: break words into wordpieces (with a separate model); example from the paper:

    `Jet makers feud over seat width with big orders at stake`

    becomes

    `_J et _makers _fe ud _over _seat _width _with _big _orders _at _stake`

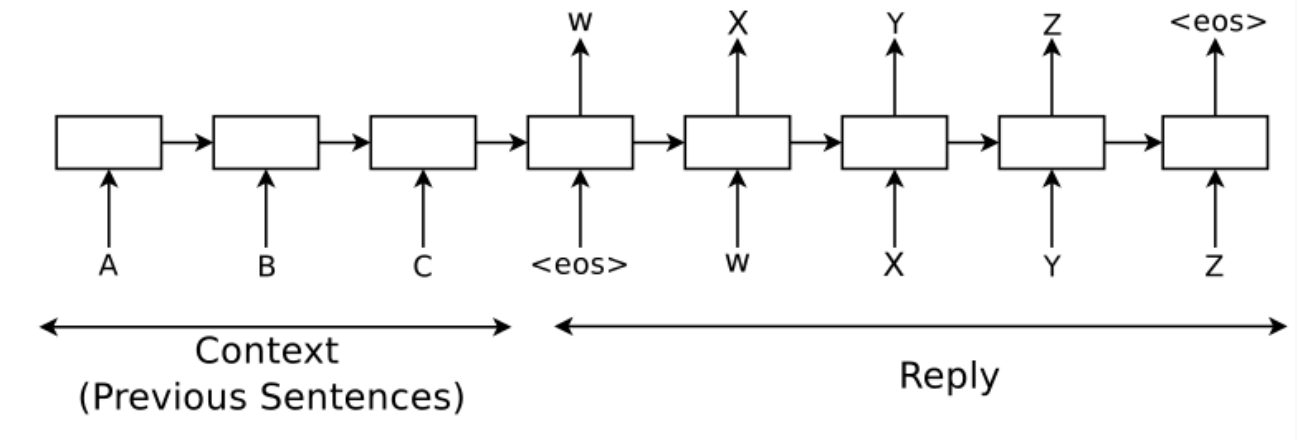
  - *mixed word/character model*: use word model but for out-of-vocabulary words convert them into characters (specifically marked so that they cannot be confused); example from the paper:

    `Miki` becomes `<B>M <M>i <M>k <E>i`

DIALOG AND CONVERSATION

DIALOG AND CONVERSATIONAL MODELS

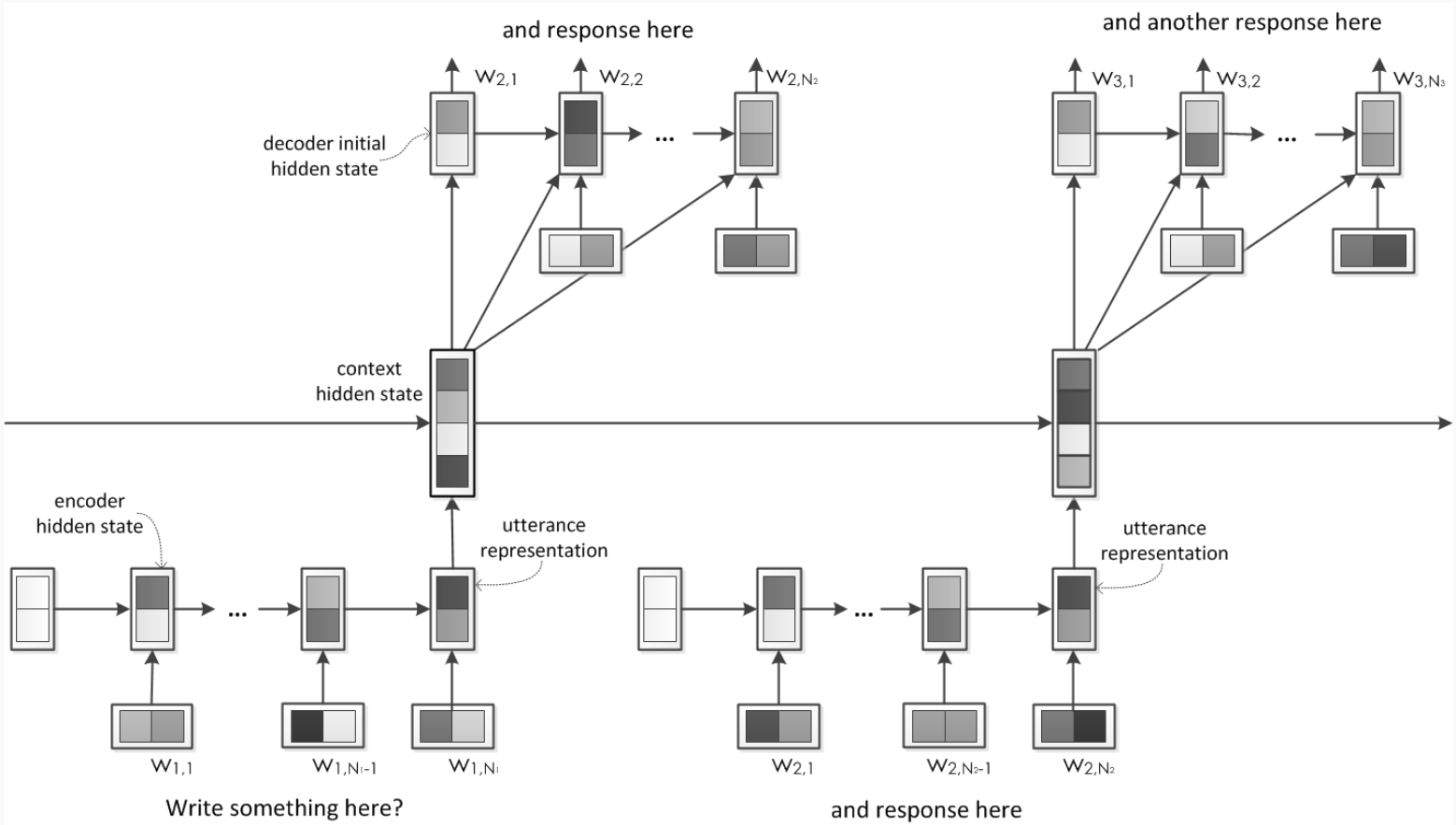
- Dialog models attempt to model and predict dialogue; conversational models actively talk to a human.
- *Not just chat bots*! Applications – automatic chat systems for business etc., so we want to convey information.
- Vinyals and Le (2015) use *seq2seq* (Sutskever et al. 2014):
  - feed previous sentences ABC as context to the RNN;
  - predict the next word of reply WXYZ based on the previous word and hidden state.



- They get a reasonable conversational model, both general (MovieSubtitles) and in a specific domain (IT helpdesk).

- Hierarchical recurrent encoder decoder architecture (HRED); first proposed for query suggestion in IR (Sordoni et al. 2015), used for dialog systems in (Serban et al. 2015).
- The dialogue as a two-level system: a sequence of utterances, each of which is in turn a sequence of words. To model this two-level system, HRED trains:
  (1) *encoder* RNN that maps each utterance in a dialogue into a single utterance vector;
  (2) *context* RNN that processes all previous utterance vectors and combines them into the current context vector;
  (3) *decoder* RNN that predicts the tokens in the next utterance, one at a time, conditional on the context RNN.

- HRED architecture:



- (Serban et al. 2015) report promising results in terms of both language models (perplexity) and expert evaluation.

- Some recent developments:
  - (Li et al., 2016a) apply, again, reinforcement learning (DQN) to improve dialogue generation;
  - (Li et al., 2016b) add *personas* with latent variables, so dialogue can be more consistent (yes, it's the same Li);
  - (Wen et al., 2016) use *snapshot learning*, adding some weak supervision in the form of particular events occurring in the output sequence (whether we still want to say something or have already said it);
  - (Su et al., 2016) improve dialogue systems with online active reward learning, a tool from reinforcement learning.
- Generally, chatbots are becoming commonplace but it is still a long way to go before actual general-purpose dialogue.

QUESTION ANSWERING

QUESTION ANSWERING

- Question answering (QA) is one of the hardest NLP challenges, close to true language understanding.
- Let us begin with evaluation:
  - it's easy to find datasets for information retrieval;
  - these questions can be answered knowledge base approaches: map questions to logical queries over a graph of facts;
  - in a multiple choice setting (Quiz Bowl), map the question and possible answers to a semantic space and find nearest neighbors (Socher et al. 2014);
  - but this is not exactly general question answering.
- (Weston et al. 2015): a dataset of simple (for humans) questions that do not require any special knowledge.
- But require reasoning and understanding of semantic structure...

QUESTION ANSWERING

- Sample questions:

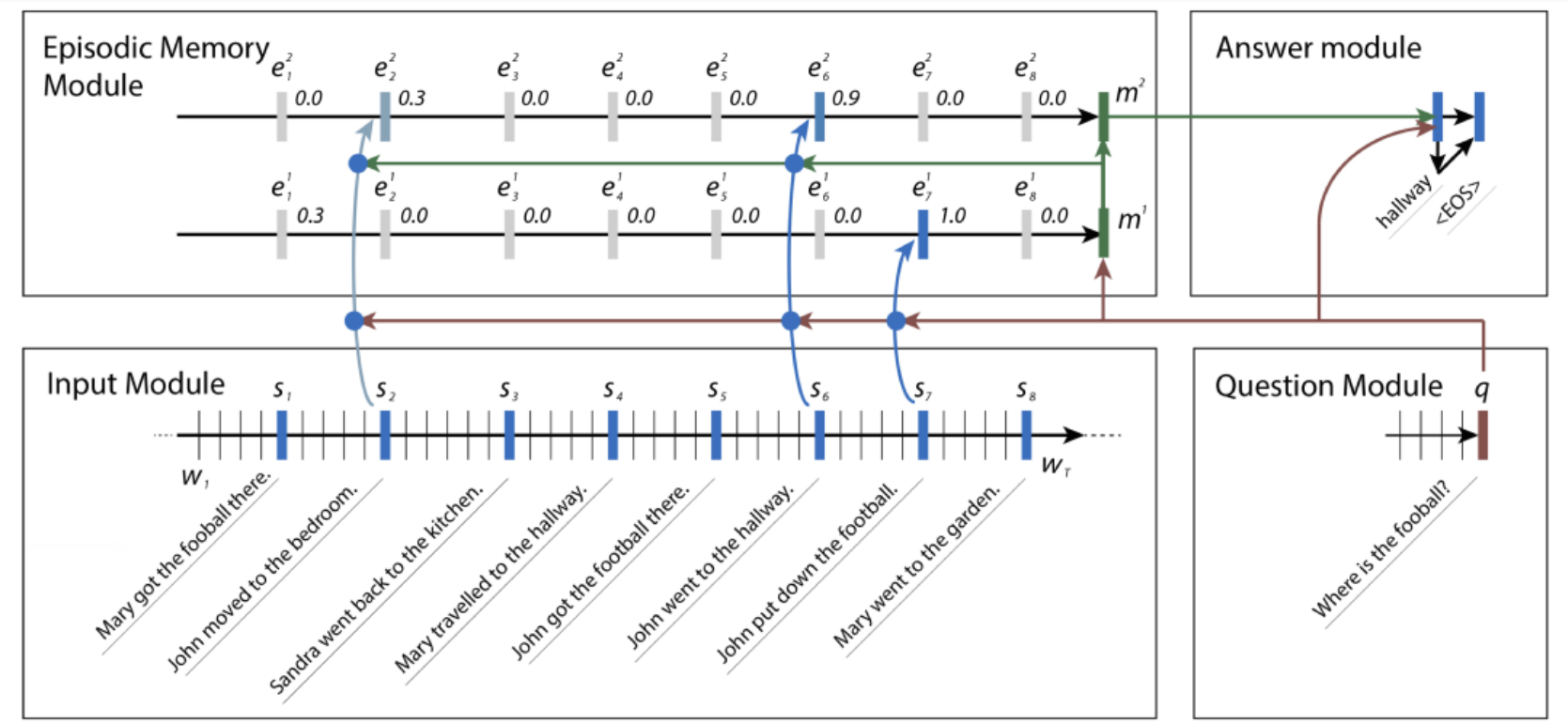
Task 1: *Single Supporting Fact* Mary went to the bathroom. John moved to the hallway. Mary travelled to the office. Where is Mary? **A**: *office*	**Task 4**: *Two Argument Relations* The office is north of the bedroom. The bedroom is north of the bathroom. The kitchen is west of the garden. What is north of the bedroom? **A**: *office* What is the bedroom north of? **A**: *bathroom*
Task 7: *Counting* Daniel picked up the football. Daniel dropped the football. Daniel got the milk. Daniel took the apple. How many objects is Daniel holding? **A**: *two*	**Task 10**: *Indefinite Knowledge* John is either in the classroom or the playground. Sandra is in the garden. Is John in the classroom? **A**: *maybe* Is John in the office? **A**: *no*
Task 15: *Basic Deduction* Sheep are afraid of wolves. Cats are afraid of dogs. Mice are afraid of cats. Gertrude is a sheep. What is Gertrude afraid of? **A**: *wolves*	**Task 20**: *Agent's Motivations* John is hungry. John goes to the kitchen. John grabbed the apple there. Daniel is hungry. Where does Daniel go? **A**: *kitchen* Why did John go to the kitchen? **A**: *hungry*

- One problem is that we have to *remember* the context set throughout the whole question...

- ...so the current state of the art are *memory networks* (Weston et al. 2014).
- An array of objects (memory) and the following components learned during training:
  - I (input feature map) converts the input to the internal feature representation;
  - G (generalization) updates old memories after receiving new input;
  - O (output feature map) produces new output given a new input and a memory state;
  - R (response) converts the output of O into the output response format (e.g., text).

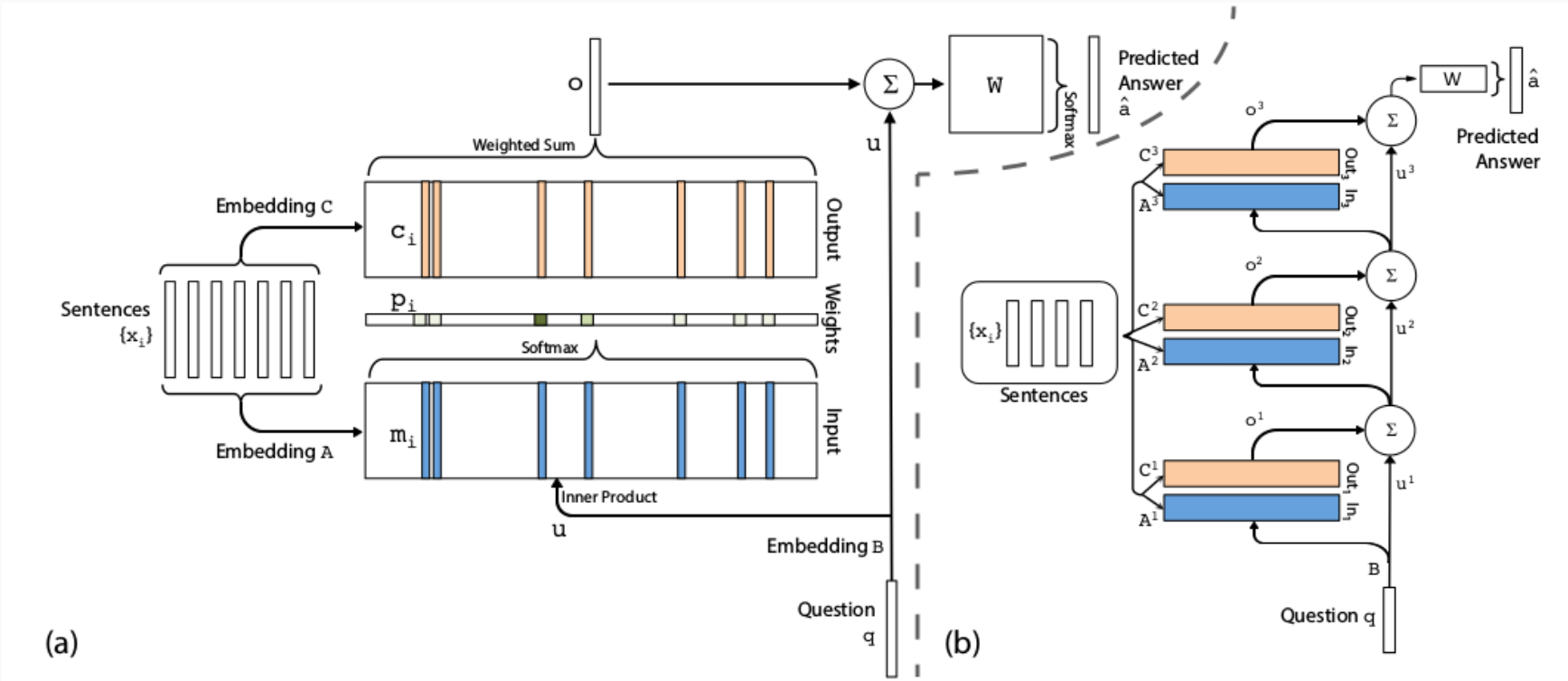
QUESTION ANSWERING

- *Dynamic memory networks* (Kumar et al. 2015).
- Episodic memory unit that chooses which parts of the input to focus on with an attention mechanism:



QUESTION ANSWERING

- *End-to-end memory networks* (Sukhbaatar et al. 2015).
- A continuous version of memory networks, with multiple hops (computational steps) per output symbol.
- Regular memory networks require supervision on each layer; end-to-end ones can be trained with input-output pairs:



QUESTION ANSWERING

- There are plenty of other extensions; one problem is how to link QA systems with knowledge bases to answer questions that require both reasoning and knowledge.
- Google DeepMind is also working on QA (Hermann et al. 2015):
  - a CNN-based approach to QA, also tested on the same dataset;
  - perhaps more importantly, a relatively simple and straightforward way to convert unlabeled corpora to questions;
  - e.g., given a newspaper article and its summary, they construct (context, query, answer) triples that could then be used for supervised training of text comprehension models.
- I expect a lot of exciting things to happen here.
- But allow me to suggest...

WHAT? WHERE? WHEN?

- «What? Where? When?»: a team game of answering questions. Sometimes it looks like this...

WHAT? WHERE? WHEN?

- ...but usually it looks like this:

what? where? when?

- Teams of $\leqslant 6$ players answer questions, whoever gets the most correct answers wins.
- db.chgk.info – database of about 300K questions.
- Some of them come from “Своя игра”, a Jeopardy clone but often with less direct questions:
  - *Современная музыка*
    На самом деле первое слово в названии **ЭТОГО** коллектива совпадает с фамилией шестнадцатого президента США, а исказили его для того, чтобы приобрести соответствующее названию доменное имя.
  - *Россия в начале XX века*
    В **ЭТОМ** году в России было собрано 5,3 миллиарда пудов зерновых.
  - *Чёрное и белое*
    ОН постоянно меняет черное на белое и наоборот, а его соседа в этом вопросе отличает постоянство.

WHAT? WHERE? WHEN?

- Most are “Что? Где? Когда?” questions, even harder for automated analysis:
  - Ягоды черники невзрачные и довольно простецкие. Какой автор утверждал, что не случайно использовал в своей книге одно из названий черники?
  - В середине тридцатых годов в Москве выходила газета под названием «Советское метро». Какие две буквы мы заменили в предыдущем предложении?
  - Русская примета рекомендует 18 июня полоть сорняки. Согласно второй части той же приметы, этот день можно считать благоприятным для **НЕЁ**. Назовите **ЕЁ** словом латинского происхождения.
  - Соблазнитель из венской оперетты считает, что после **НЕГО** женская неприступность уменьшается вчетверо. Назовите **ЕГО** одним словом.
- I believe it is a great and very challenging QA dataset.
- How far in the future do you think it is? :)

THANK YOU!

Thank you for your attention!