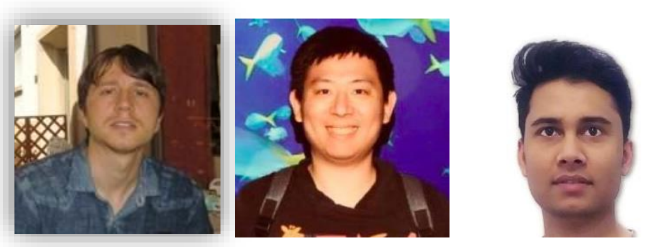
Deep Learning for NLP

Lecture 6 – Word Embeddings 2 (Syntactic, Bilingual, Contextualized Embeddings)

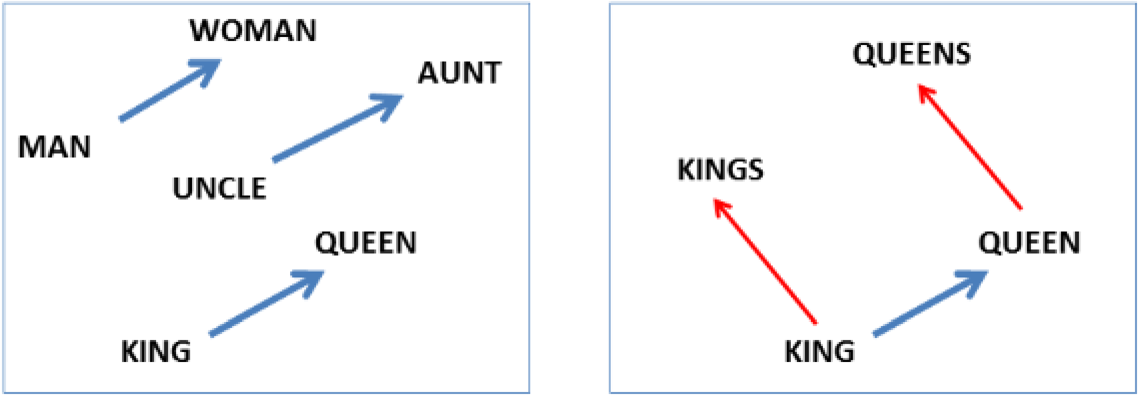
Dr. Steffen Eger
Wei Zhao
Niraj Pandey



Natural Language Learning Group (NLLG)
Technische Universität Darmstadt

Last session

- Word embeddings can represent semantic and syntactic relations between words in the vector space



Mikolov et al (2013a)

Linguistic Regularities in Continuous Space Word Representations

This lecture

1) Multi-Lingual, Multi-Sense Word Embeddings

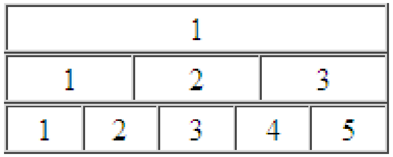
2) Syntactic Word Embeddings

3) Miscellaneous

www.shutterstock.com • 361339859

Word Senses

- Words do not represent only one meaning



1

1 2 3

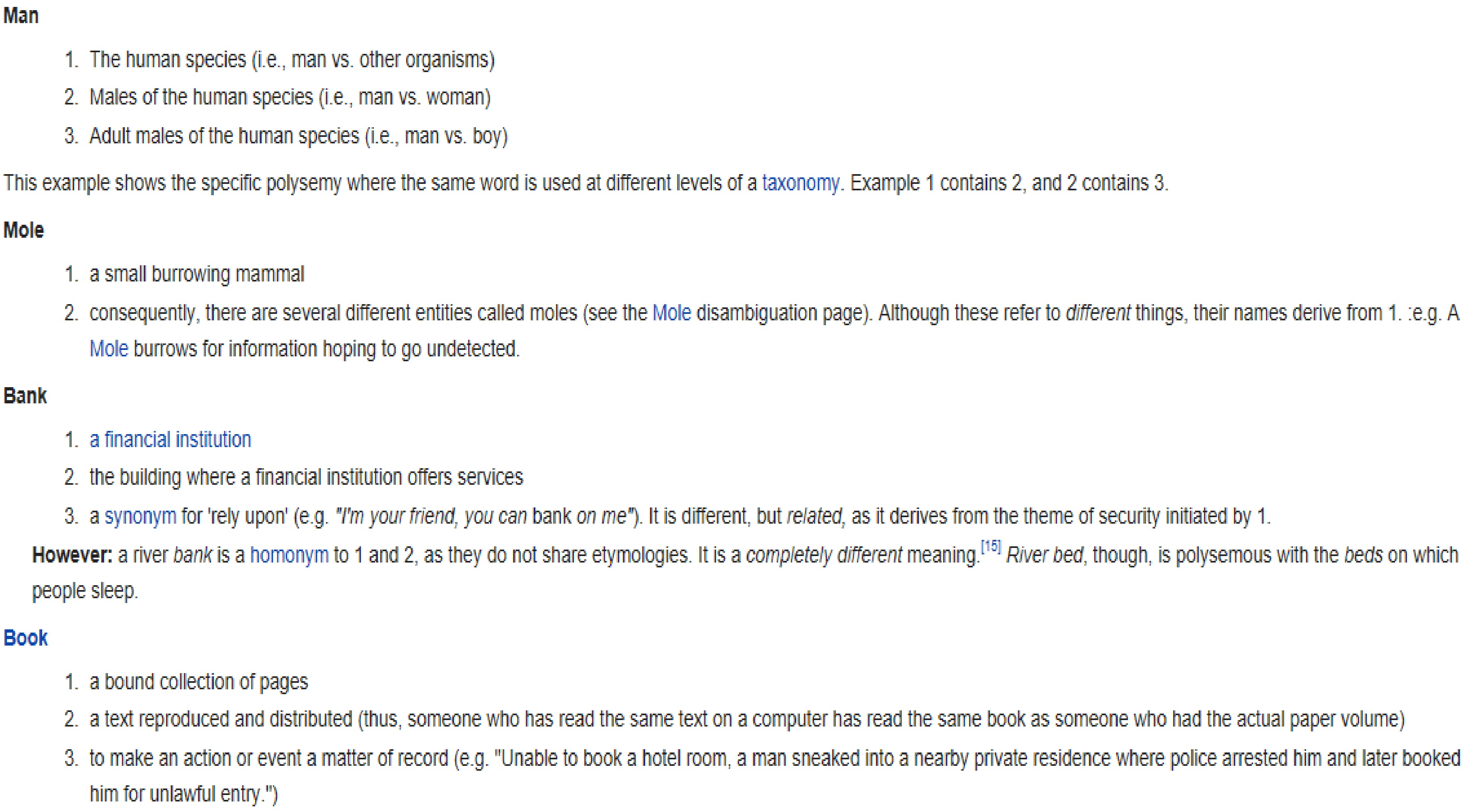
1 2 3 4 5

…

You are pretty fly…

- Problem is generally known as *polysemy* (or even *homonymy*): a word may have many different meanings:
  - bank, table, fly, man, …

Word Senses



Man

1. The human species (i.e., man vs. other organisms)
2. Males of the human species (i.e., man vs. woman)
3. Adult males of the human species (i.e., man vs. boy)

This example shows the specific polysemy where the same word is used at different levels of a taxonomy. Example 1 contains 2, and 2 contains 3.

Mole

1. a small burrowing mammal
2. consequently, there are several different entities called moles (see the Mole disambiguation page). Although these refer to *different* things, their names derive from 1. :e.g. A Mole burrows for information hoping to go undetected.

Bank

1. a financial institution
2. the building where a financial institution offers services
3. a synonym for 'rely upon' (e.g. *"I'm your friend, you can* bank *on me"*). It is different, but *related,* as it derives from the theme of security initiated by 1.

However: a river *bank* is a homonym to 1 and 2, as they do not share etymologies. It is a *completely different* meaning.[15] *River bed*, though, is polysemous with the *beds* on which people sleep.

Book

1. a bound collection of pages
2. a text reproduced and distributed (thus, someone who has read the same text on a computer has read the same book as someone who had the actual paper volume)
3. to make an action or event a matter of record (e.g. "Unable to book a hotel room, a man sneaked into a nearby private residence where police arrested him and later booked him for unlawful entry.")

Sense-disambiguated word representations

- Idea: Train word vectors on sense-disambiguated corpora

  Example from the SemCor corpus:

```
<s snum=132>
<wf cmd=ignore pos=DT>A</wf>
<wf cmd=done pos=NN lemma=rush wnsn=2 lexsn=1:11:00::>rush</wf>
<wf cmd=ignore pos=IN>of</wf>
<wf cmd=done pos=NN lemma=panic wnsn=1 lexsn=1:12:00::>panic</wf>
<wf cmd=done pos=VB lemma=catch wnsn=12 lexsn=2:30:00::>caught</wf>
<wf cmd=done rdf=person pos=NNP lemma=person wnsn=1 lexsn=1:03:00::
pn=person>Sarah</wf>
<punc>.</punc>
</s>
```

Sense-disambiguated word representations

- Idea: Train word vectors on sense-disambiguated corpora

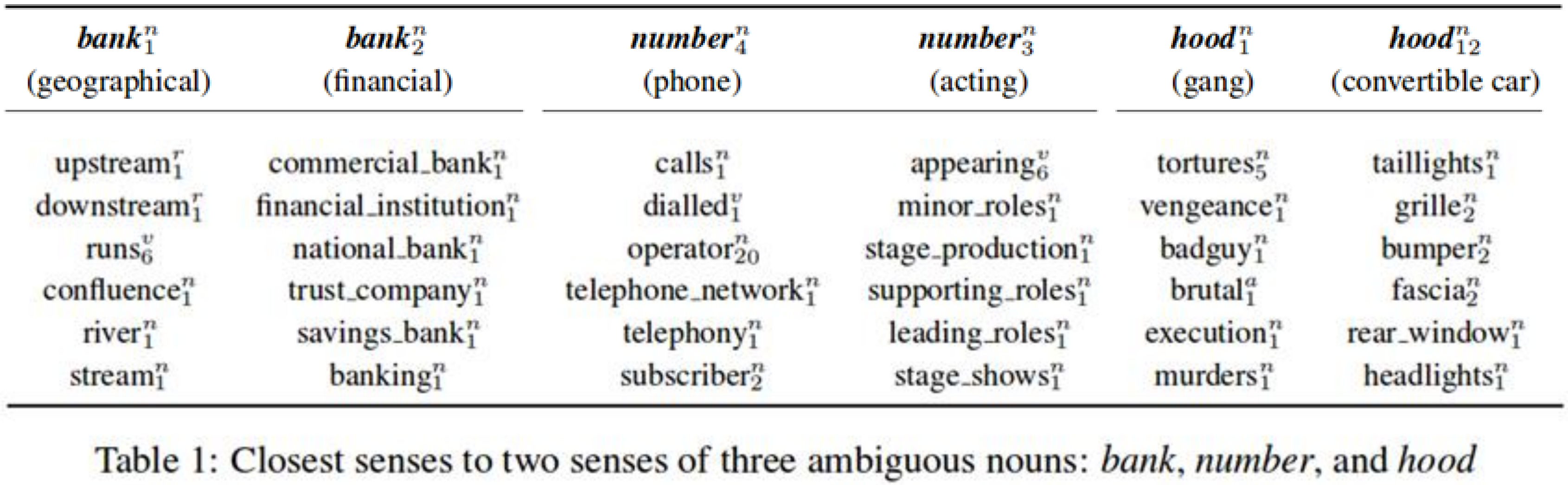
  Example from the SemCor corpus:

```
<s snum=132>
<wf cmd=ignore pos=DT>A</wf>
<wf cmd=done pos=NN lemma=rush wnsn=2 lexsn=1:11:00::>rush</wf>
<wf cmd=ignore pos=IN>of</wf>
<wf cmd=done pos=NN lemma=panic wnsn=1 lexsn=1:12:00::>panic</wf>
<wf cmd=done pos=VB lemma=catch wnsn=12 lexsn=2:30:00::>caught</wf>
<wf cmd=done rdf=person pos=NNP lemma=person wnsn=1 lexsn=1:03:00::
pn=person>Sarah</wf>
<punc>.</punc>
</s>
```

→ A rush_2 of panic_1 caught_12 Sarah_1

Sense-disambiguated word representations

- Result: different representations for each sense



bank$^n_1$ (geographical)	***bank***$^n_2$ (financial)	***number***$^n_4$ (phone)	***number***$^n_3$ (acting)	***hood***$^n_1$ (gang)	***hood***$^n_{12}$ (convertible car)
upstream$^r_1$	commercial_bank$^n_1$	calls$^n_1$	appearing$^v_6$	tortures$^n_5$	taillights$^n_1$
downstream$^r_1$	financial_institution$^n_1$	dialled$^v_1$	minor_roles$^n_1$	vengeance$^n_1$	grille$^n_2$
runs$^v_6$	national_bank$^n_1$	operator$^n_{20}$	stage_production$^n_1$	badguy$^n_1$	bumper$^n_2$
confluence$^n_1$	trust_company$^n_1$	telephone_network$^n_1$	supporting_roles$^n_1$	brutal$^a_1$	fascia$^n_2$
river$^n_1$	savings_bank$^n_1$	telephony$^n_1$	leading_roles$^n_1$	execution$^n_1$	rear_window$^n_1$
stream$^n_1$	banking$^n_1$	subscriber$^n_2$	stage_shows$^n_1$	murders$^n_1$	headlights$^n_1$

Table 1: Closest senses to two senses of three ambiguous nouns: *bank*, *number*, and *hood*

- Iacobacci et al (2015): *SensEmbed: Learning Sense Embeddings for Word and Relational Similarity*

Problems

- How do you now train an NLP system with these sense-disambiguated embeddings?

A more parsimonious approach

- Run word2vec on your data and compute embeddings

- For each target word, represent its context as avg. or concatenated embedding
  - ... need to go to the bank to get some money ….
  - … debt by utilizing a credit line granted by a bank …
  - …. raw water is largely river bank filtrate (approximately 70 percent) …
  - … runs from its idyllic river bank promenade under the Elbe to …

A more parsimonious approach

- Run word2vec on your data and compute embeddings

- For each target word, represent its context as avg. or concatenated embedding
    - ... need to go to the bank to get some money ….
    - … debt by utilizing a credit line granted by a bank …
    - …. raw water is largely river bank filtrate (approximately 70 percent) …
    - … runs from its idyllic river bank promenade under the Elbe to …

A more parsimonious approach

- Run word2vec on your data and compute embeddings
- For each target word, represent its context as avg. or concatenated embedding
  - ... need to go to the bank to get some money ….

    context = [.2,.8]
  - … debt by utilizing a credit line granted by a bank …

    context = [.4,.6]
  - …. raw water is largely river bank filtrate (approximately 70 percent) …

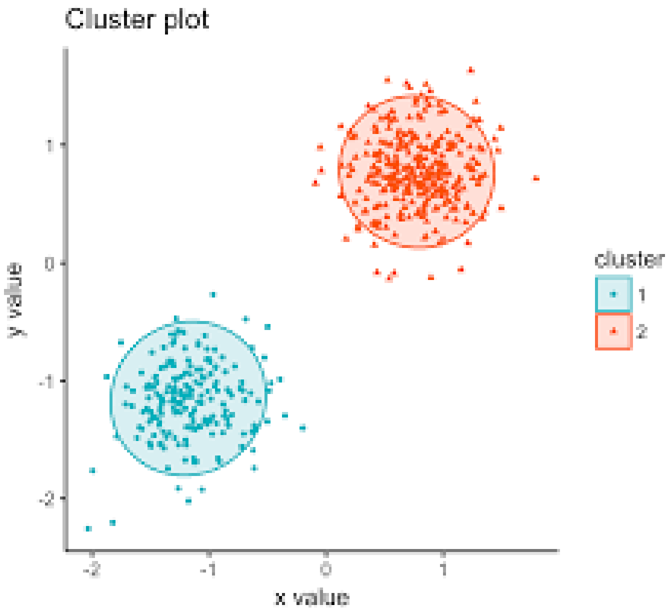
    context = [-.2,-.8]
  - … runs from its idyllic river bank promenade under the Elbe to …

    context = [-.9,-.3]
- Cluster the context representations, and assign each word's context to a cluster → the word has the sense corresponding to the cluster index
  - Using techniques from *unsupervised* machine learning (see lecture 2)
- Run word2vec on sense-disambiguated corpus

A more parsimonious approach

- Run word2vec on your data and compute embeddings

- For each target word, represent its context as avg embedding
  - ... need to go to the bank to get some mo
  - … debt by utilizing a credit line granted by
  - …. raw water is largely river bank filtrate percent) …
  - … runs from its idyllic river bank promena



- Cluster the context representations, and assign each word's context to a cluster → the word has the sense corresponding to the cluster index
  - Using techniques from *unsupervised* machine learning (see lecture 2)
- Run word2vec on sense-disambiguated corpus

Sense-disambiguated word representations

However, in practice, most people didn’t use sense embeddings

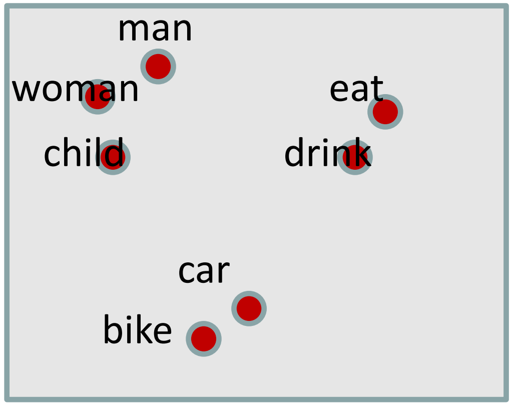
- Not so much benefit in using them in practical applications
- On the other hand, the cost is much higher --- one needs a sense-labeler or a more complicated model

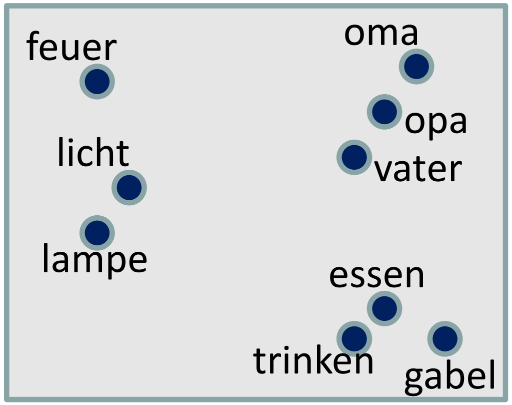
- Before ELMo and BERT came around in 2018 (see below) …
  - With **contextualized word embeddings**

Bilingual Embeddings

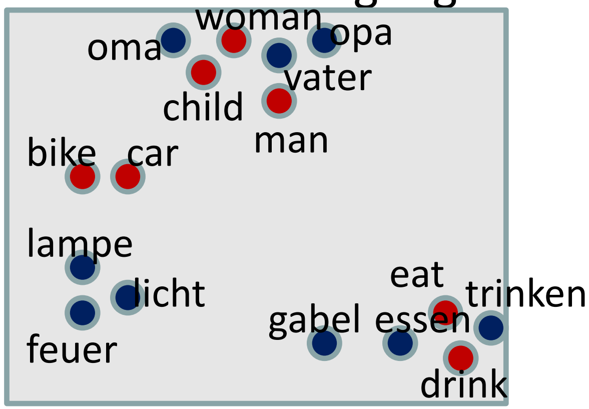
- Word representations for two languages:

  → train on corpus from both languages



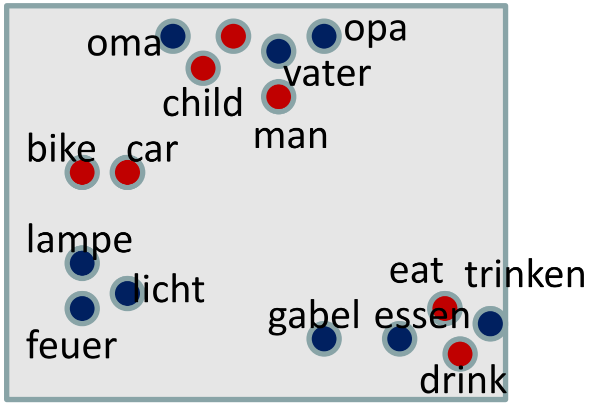


- Goal: represent words from different languages in the same space



Bilingual Embeddings

Goal: represent words from different languages in the same space



Bilingual Embeddings – General idea

- Can think of it as having two objectives we want to satisfy

- **cross-lingual objective:** words that are translations of each other should be close in the projected space
- **mono-lingual objective:** words that occur in monolingually similar contexts should be close to each other in vector space

Bilinguality – Why?

(1) Second language may act as an additional "signal"

- Which may help to improve word embeddings even in the first language
  - → **Make Monolingual Embeddings better**

- E.g. assume that some word like "opa" occurs very infrequently in the German corpus, thus it's difficult to reliably estimate its word embedding
- If its English translation "grandfather" occurs frequently in the English corpus, the German word should get a more appropriate embedding in the bilingual space

Bilinguality – Why?

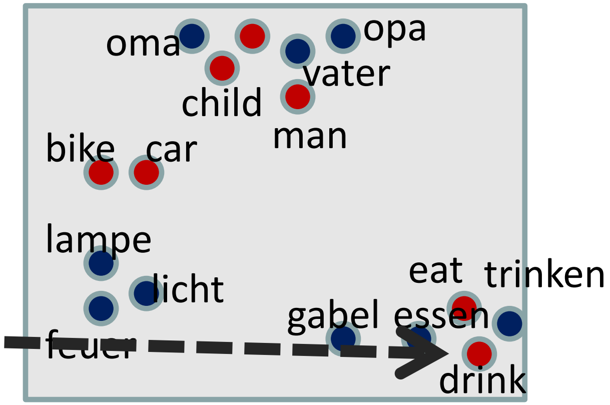
(2) If words are projected in a common space ("shared features"), this may allow for **Direct Transfer**

- Train a model in one language (usually resource-rich)
- Directly apply in another language (usually resource-poor)

Bilinguality – Why?

(2) Example Direct Transfer: task is POS tagging

- Goal / approach:
  - *Train:* I may not drink this → PRON VERB PARTICLE VERB DET
  - *Test*: Es ist wichtig, ausreichend zu trinken → ....
- Training (idea):
  - Input: center words with their context words
  - Output: labels of center word
  - E.g. (not,drink,this) → VERB

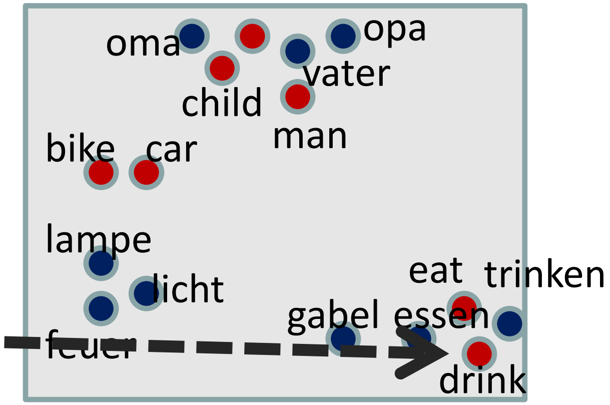


- **Direct transfer aka zero-shot transfer:**
  - train using bilingual embeddings in English (assume big labeled English dataset)
  - then apply to German data
- Problems with the Direct Transfer approach?

Bilinguality – Why?

(2) Example Direct Transfer: task is POS tagging

- Goal / approach:
  - *Train:* I may not drink this → PRON VERB PARTICLE VERB DET
  - *Test*: Es ist wichtig, ausreichend zu trinken → ….
- Training (idea):
  - Input: center words with their context words
  - Output: labels of center word
  - E.g. (not,drink,this) → VERB



- Direct transfer:
  - train using bilingual embeddings in English (assume big labeled English dataset)
  - then apply to German data
- Problems with the Direct Transfer approach?
  - "OOV words", syntactic ordering

Bilingual Embeddings – Naive Approach

- Given 1: Monolingual Embeddings (e.g. English, German)
- Given 2: Dictionary EN ← → DE

- Translate German words to English words, assign them the embedding of the English word (or concatenate, average, …)
  - Bottleneck is the dictionary
  - Cannot assign meanings to words that are not in the dictionary

Bilingual Embeddings

- More sophisticated approaches have been suggested, relying on different kinds of (costly) information

Approach 1: Learning a transformation matrix

- One of the first and simplest approaches

  Mikolov et al. 2013, Exploiting similarities among languages for machine translation

- Given: monolingual embeddings + dictionary
  - Dictionary: *cat-Katze*, *table-Tisch*, …

x_i	z_i
cat	Katze
table	Tisch
…	…

Approach 1: Learning a transformation matrix

- One of the first and simplest approaches

  Mikolov et al. 2013, Exploiting similarities among languages for machine translation

- Given: monolingual embeddings + dictionary
  - Dictionary: *cat-Katze*, *table-Tisch*, …

x_i	z_i
[0.2,-0.3,0.8]	[0.5,0.9,-1]
[1,2,-5]	[0.1,-0.1,0.1]
…	…

Approach 1: Learning a transformation matrix

- One of the first and simplest approaches

  Mikolov et al. 2013, Exploiting similarities among languages for machine translation

- Given: monolingual embeddings + dictionary
  - Dictionary: *cat-Katze*, *table-Tisch*, …
- We estimate a linear transformation from this data:
  - $\min_{\boldsymbol{W}} \sum_i ||\boldsymbol{x}_i \boldsymbol{W} - \boldsymbol{z}_i||^2$
  - $\boldsymbol{x}_i$ and $\boldsymbol{z}_i$ are monolingual vectors of words from dictionary

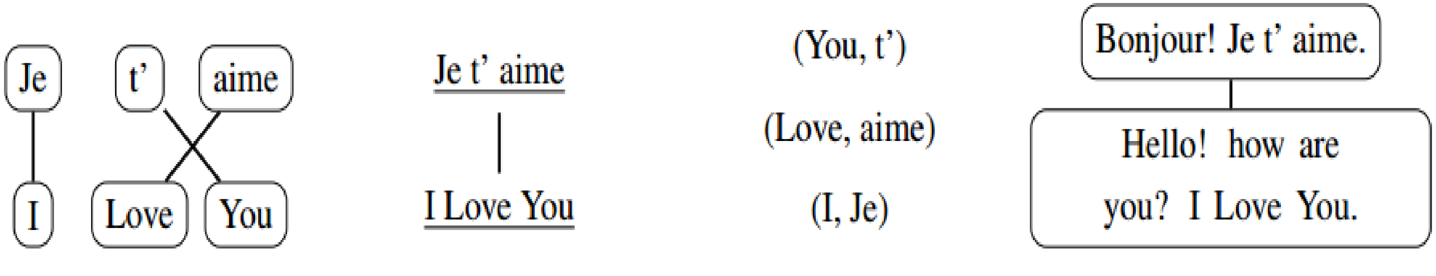
How to solve this?

- Once $\boldsymbol{W}$ is learned, we can map any language x word into the space of language z
  - Even words for which we do not have translations

More Bilingual Embeddings

- See Upadhayay et al. (2016), Cross-lingual Models of Word Embeddings: An Empirical Comparison
- And more recent Glavas et al. (2019), How to (properly) evaluate cross-lingual word embeddings: On Strong Baselines, Comparative Analyses, and Some Misconceptions
  - for survey papers

More Bilingual Embeddings



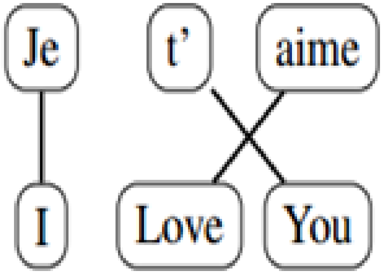
(a) **BiSkip** (b) **BiCVM** (c) **BiCCA** (d) **BiVCD**

Figure 2: Forms of supervision required by the four models compared in this paper. From left to right, the cost of the supervision required varies from expensive (BiSkip) to cheap (BiVCD). BiSkip requires a parallel corpus annotated with word alignments (Fig. 2a), BiCVM requires a sentence-aligned corpus (Fig. 2b), BiCCA only requires a bilingual lexicon (Fig. 2c) and BiVCD requires comparable documents (Fig. 2d).

Bilingual Embeddings

- We discuss (a) BiSkip and (d) BiVCD

- **BiSkip** uses sentence and word aligned texts, then runs a skip-gram model whose contexts are words from both languages:
  - E.g. on input *love* BiSkip wants to predict the context *je, I, you, t'*;
  - similar for *aime: t', you*
  - → similar contexts are predicted → similar representations

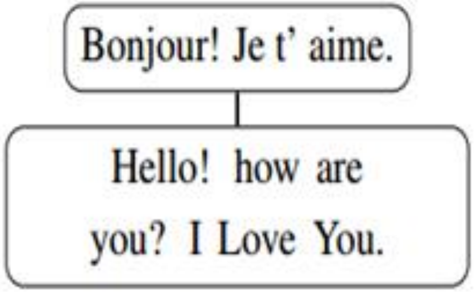


(a) **BiSkip**

Bilingual Embeddings

- We discuss (a) BiSkip and (d) BiVCD

- **BiVCD** is even simpler. Given aligned documents (e.g. Wikipedia articles)
  - Merge them, then random shuffle all words
  - Then run a Monolingual Model (e.g. CBOW, Glove, Skip-Gram) on it
  - Why does this yield meaningful results?



(d) **BiVCD**

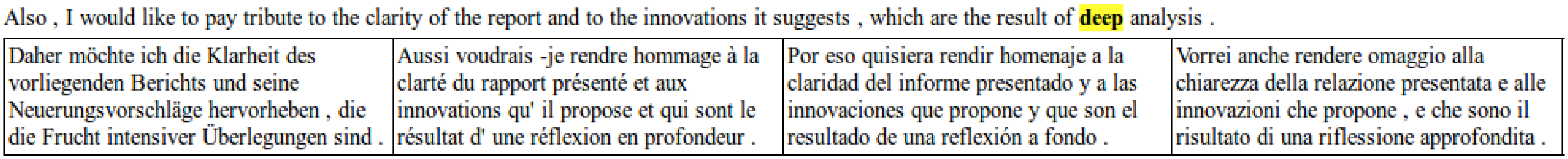
Determining bi-lingual mappings (for BiSkip)

- Dictionary
- Inter-lingual links in Wikipedia
- Word alignments learned from parallel corpora

Determining bi-lingual mappings

- Dictionary
- Inter-lingual links in Wikipedia
- Word/Sentence alignments learned from parallel corpora

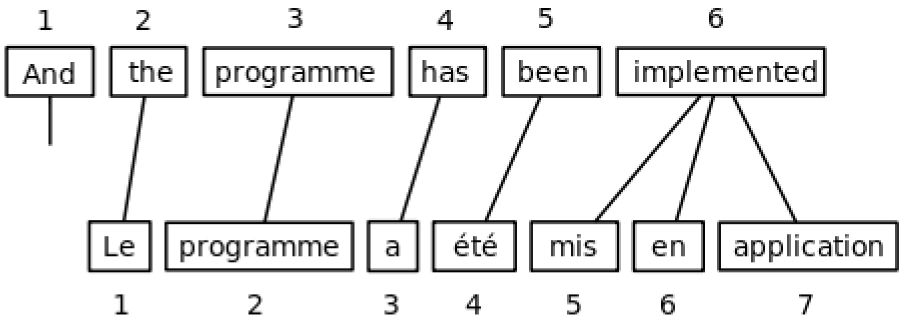
Europarl: parallel corpus from the European parliament visualized by IMS:



Also , I would like to pay tribute to the clarity of the report and to the innovations it suggests , which are the result of **deep** analysis .

Daher möchte ich die Klarheit des vorliegenden Berichts und seine Neuerungsvorschläge hervorheben , die die Frucht intensiver Überlegungen sind .	Aussi voudrais -je rendre hommage à la clarté du rapport présenté et aux innovations qu' il propose et qui sont le résultat d' une réflexion en profondeur .	Por eso quisiera rendir homenaje a la claridad del informe presentado y a las innovaciones que propone y que son el resultado de una reflexión a fondo .	Vorrei anche rendere omaggio alla chiarezza della relazione presentata e alle innovazioni che propone , e che sono il risultato di una riflessione approfondita .

Learn word alignments



Multilinguality

- We talked about mapping two languages in a common space

- How about 3, 5, 10 languages?

- Much less explored topic

- However, there is work on it, such as Ammar et al. (2016), Massively Multilingual word embeddings

  - They extend BiCCA to MultiCCA and BiSkip to MultiSkip

- In recent years, people use **Multilingual BERT** (MBERT), which yields embeddings in a joint space for 100+ languages

Current trends

- Learn bilingual word embeddings from as few resources as possible,
  - E.g., only 10 aligned word pairs (can be punctuation)

- E.g. Artexte et al., Learning bilingual word embeddings with (almost) no bilingual data, ACL 2017

  - From there we can go to unsupervised machine translation
    - Without any parallel data (crazy stuff!)

Current trends

- E.g. Artexte et al., Learning bilingual word embeddings with (almost) no bilingual data, ACL 2017
- Main idea:
  - If we had a dictionary, we can get bilingual embeddings
  - If we had bilingual embeddings, we can get a dictionary

Current trends

- E.g. Artexte et al., Learning bilingual word embeddings with (almost) no bilingual data, ACL 2017
- Idea:
  - 1) Use a lexicon (*seed lexicon* is easy to get automatically)
  - 2) Learn bilingual embeddings using current lexicon (→ Mikolov’s method, i.e., “Approach 1”)
  - 3) Get a better lexicon using bilingual embeddings
  - 4) Go back to 1)

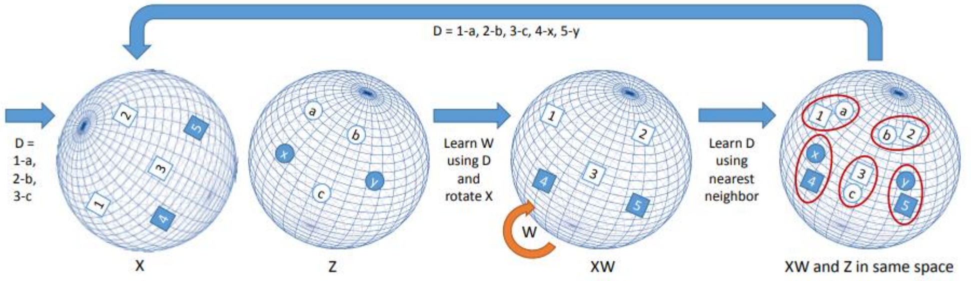


Figure 1: A general schema of the proposed self-learning framework. Previous works learn a mapping W based on the seed dictionary D, which is then used to learn the full dictionary. In our proposal we use the new dictionary to learn a new mapping, iterating until convergence.

This lecture

1) Multi-Sense and Multi-Lingual Word Embeddings

2) Syntactic Word Embeddings

3) Miscellaneous

More syntactically oriented embeddings

- **Syntactic** relations between words should also be represented in the vectors

→ Problem: word order matters

Dog bites man. vs Man bites dog.

Position Information

- Remember: The word2vec models do not consider position information:
  - No distinction between left and right context
  - No distinction between close and far contexts

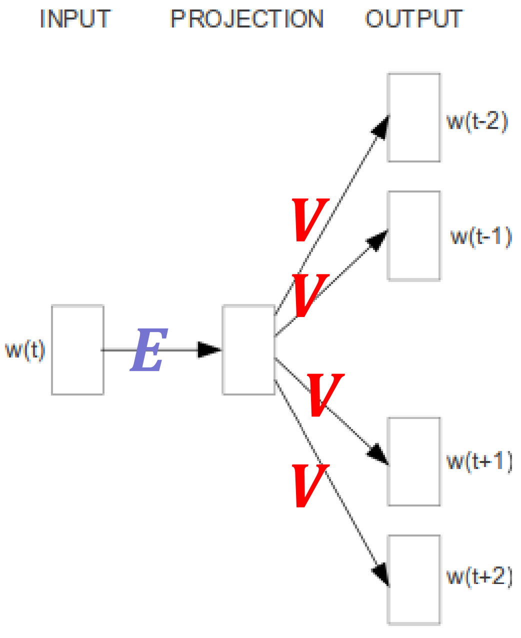
Skip-gram: ___ *bites* ___

→ *(bites, man) , (bites, dog)*

- dog bites man vs man bites dog
  - *(bites, dog-1), (bites, man+1) vs (bites, man-1), (bites, dog+1)*

- This is “intuitively” what we want (although we don’t add indices to words; why?)

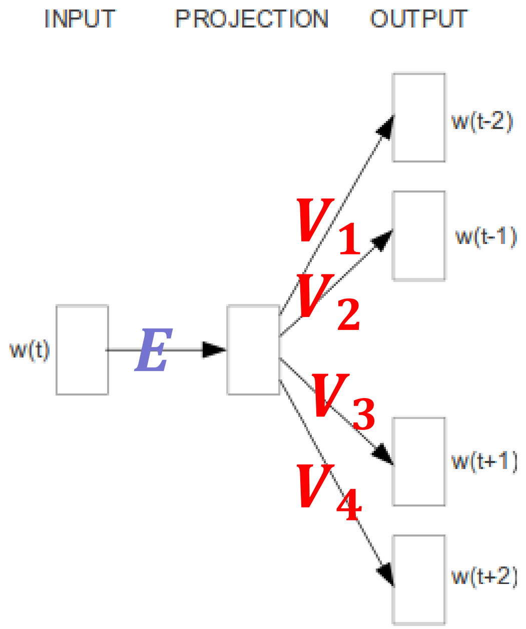
The Skip-gram model



Skip-gram

How can we predict different words when V is always the same?

The Structured Skip-gram model



Structured Skip-gram

Results

- Nearest neighbours for “*breaking*”

Skip-gram	Structured Skip-gram
breaks	*putting*
turning	*turning*
broke	*sticking*
break	*pulling*
stumbled	*picking*

- Word representations with positional information work slightly better for syntactic tasks like POS-tagging and parsing.

- Ling et al. 2015: *Two/Too Simple Adaptations of Word2Vec for Syntax Problems*

Long-distance dependencies

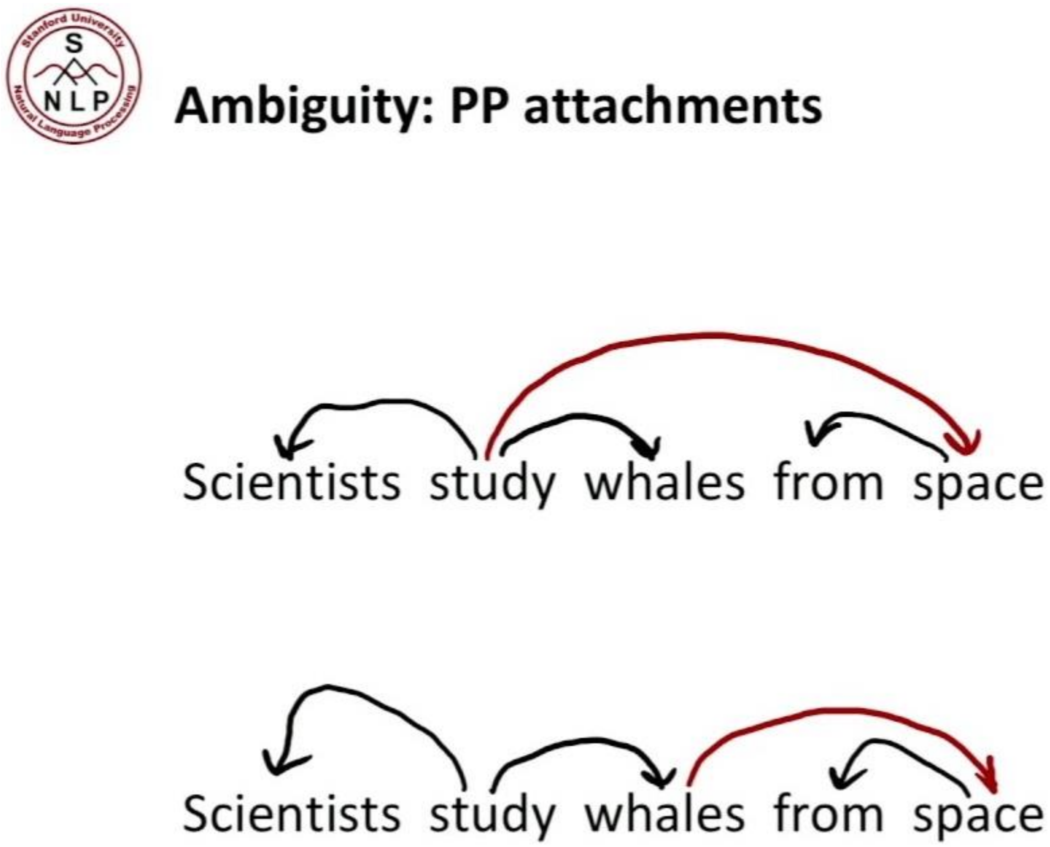
- Words can be similar with respect to verb selection preferences
  - tea/milk/beer/coffee can all be an object of the verb *drink*

- Words that share syntactic relations might be distant in a sentence:

*I would like to **drink** a very hot tall decaf half-soy (…) white chocolate **mocha***

Dependency parsing in one slide

- Outlines grammatical **relationships** between words in a sentence



Ambiguity: PP attachments

Scientists study whales from space

Scientists study whales from space

Dependency parses

- Idea: apply dependency parsing first

I would like to ***drink*** *a very hot tall decaf half-soy (…) white chocolate* ***mocha***

Output of Stanford dependency parser:

nsubj(like-3, I-1) nsubj(drink-5, I-1) aux(like-3, would-2)
root(ROOT-0, like-3) mark(drink-5, to-4) xcomp(like-3, drink-5)
det(mocha-14, a-6) advmod(hot-8, very-7) amod(mocha-14, hot-8)
amod(mocha-14, tall-9) amod(mocha-14, decaf-10) amod(mocha-14, half-soy-11)
amod(mocha-14, white-12) compound(mocha-14, chocolate-13)
dobj(drink-5, mocha-14)

Dependency-based embeddings

I would like to drink a very hot tall decaf half-soy (…) white chocolate mocha

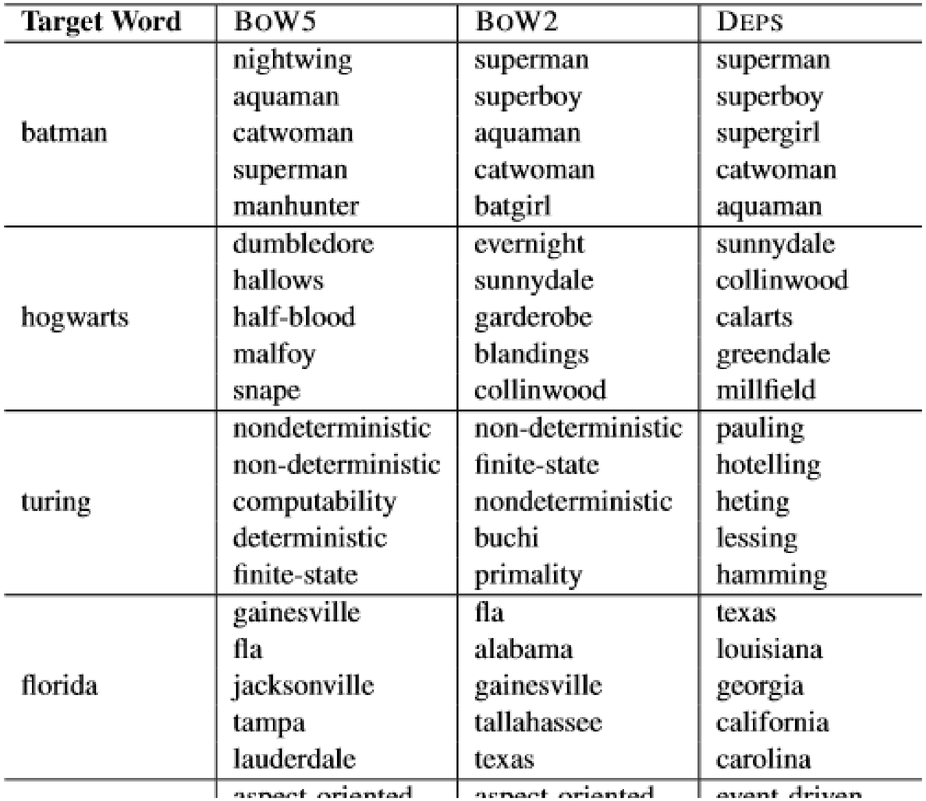
nsubj(like-3, I-1)
nsubj(drink-5, I-1)
aux(like-3, would-2)
root(ROOT-0, like-3)
mark(drink-5, to-4)
xcomp(like-3, drink-5)
det(mocha-14, a-6)
advmod(hot-8, very-7)
amod(mocha-14, hot-8)
amod(mocha-14, tall-9)
amod(mocha-14, decaf-10)
amod(mocha-14, half-soy-11)
amod(mocha-14, white-12)
compound(mocha-14, chocolate-13)
dobj(drink-5, mocha-14)

- Levy and Goldberg, 2014: *Dependency-Based Word Embeddings*

Word	Dependency Context
like	I/nsubj, would/aux, drink/xcomp
drink	I/nsubj, to/mark, mocha/dobj, like/xcomp^{-1}
hot	very/advmod, mocha/amod^{-1}
…	…

Dependency-based embeddings

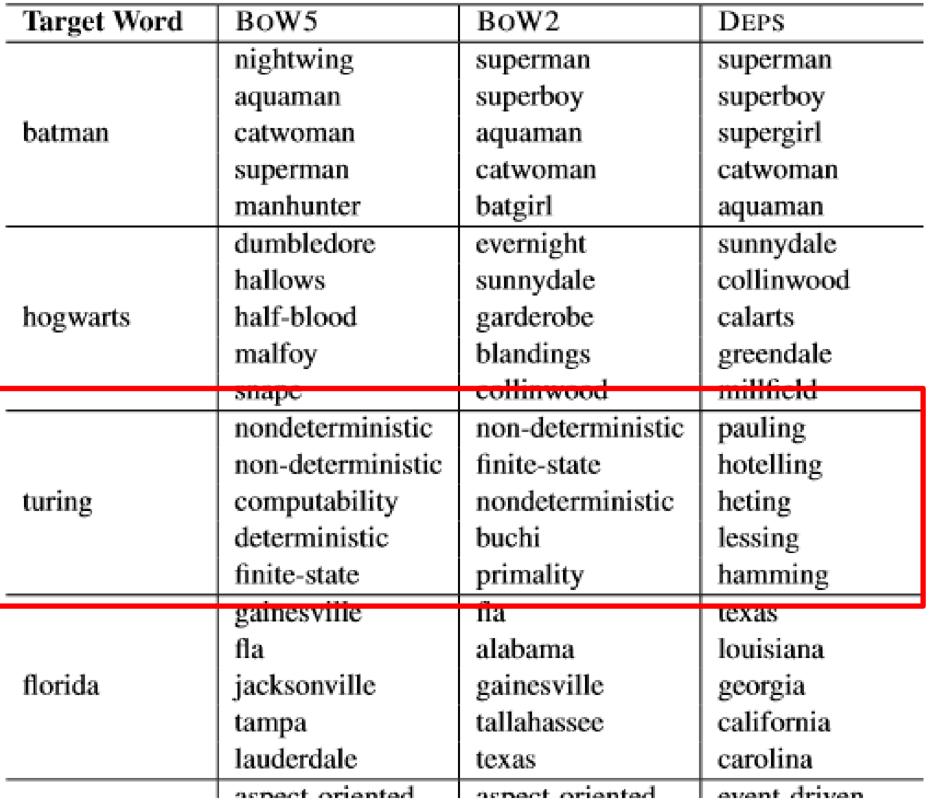
- Word2Vec finds words that **associate with** other words, while DepEmbeddings finds words **behave like** others
  - *Domain similarity* vs. *functional similarity*



Target Word	BoW5	BoW2	Deps
batman	nightwing aquaman catwoman superman manhunter	superman superboy aquaman catwoman batgirl	superman superboy supergirl catwoman aquaman
hogwarts	dumbledore hallows half-blood malfoy snape	evernight sunnydale garderobe blandings collinwood	sunnydale collinwood calarts greendale millfield
turing	nondeterministic non-deterministic computability deterministic finite-state	non-deterministic finite-state nondeterministic buchi primality	pauling hotelling heting lessing hamming
florida	gainesville fla jacksonville tampa lauderdale	fla alabama gainesville tallahassee texas	texas louisiana georgia california carolina
	aspect-oriented	aspect-oriented	event-driven

Dependency-based embeddings

- Word2Vec finds words that **associate with** other words, while DepEmbeddings finds words **behave like** others
  - *Domain similarity* vs. *functional similarity*



Target Word	BoW5	BoW2	Deps
batman	nightwing aquaman catwoman superman manhunter	superman superboy aquaman catwoman batgirl	superman superboy supergirl catwoman aquaman
hogwarts	dumbledore hallows half-blood malfoy snape	evernight sunnydale garderobe blandings collinwood	sunnydale collinwood calarts greendale millfield
turing	nondeterministic non-deterministic computability deterministic finite-state	non-deterministic finite-state nondeterministic buchi primality	pauling hotelling heting lessing hamming
florida	gainesville fla jacksonville tampa lauderdale	fla alabama gainesville tallahassee texas	texas louisiana georgia california carolina
	aspect-oriented	aspect-oriented	event-driven

This lecture

1) Multi-Sense and Multi-Lingual Word Embeddings

2) Syntactic Word Embeddings

3) **Miscellaneous**

Embeddings and Lexical Resources

- Many NLP researchers have proposed to combine NLP (linguistic) resources (which e.g. capture meaning) with the now classical word vectors
  - Faruqui et al. (2015) combine resources such a the paraphrase database (PPDB) with Embeddings
    - PPDB lists synonyms, extracted from bi-lingual datasets
    - Their model:

$$\Psi(Q) = \sum_{i=1}^{n}\left[\alpha_i\|q_i - \hat{q}_i\|^2 + \sum_{(i,j)\in E}\beta_{ij}\|q_i - q_j\|^2\right]$$

Embeddings and Lexical Resources

- Many NLP researchers have proposed to combine NLP (linguistic) resources (which e.g. capture meaning) with the now classical word vectors
  - Faruqui et al. (2015) combine resources such a the paraphrase database (PPDB) with Embeddings
    - PPDB lists synonyms, extracted from bi-lingual datasets
    - Their model:

Original word vector

$$\Psi(Q) = \sum_{i=1}^{n} \left[\alpha_i \|q_i - \hat{q}_i\|^2 + \sum_{(i,j)\in E} \beta_{ij} \|q_i - q_j\|^2 \right]$$

Embeddings and Lexical Resources

- Many NLP researchers have proposed to combine NLP (linguistic) resources (which e.g. capture meaning) with the now classical word vectors

  - Faruqui et al. (2015) combine resources such a the paraphrase database (PPDB) with Embeddings
    - PPDB lists synonyms, extracted from bi-lingual datasets
    - Their model:

New word vector

$$\Psi(Q) = \sum_{i=1}^{n} \left[\alpha_i \| q_i - \hat{q}_i \|^2 + \sum_{(i,j)\in E} \beta_{ij} \| q_i - q_j \|^2 \right]$$

Embeddings of other things than words

- Embed other stuff than words:
  - **Characters**: *i n s i g h t f u l*
    - However, there are no pre-trained embeddings on the net, why?
  - Or **syllables**: *in + sight + ful*
  - Or **morphemes**:
    - *insightful = insight + ful*
    - *helping = help + ing*
    - *greedily = greedy + ly*
    - *Dampfschifffahrt = Dampf+Schiff+Fahrt*
      - Useful (?) particulary for morphologically rich languages like
        - German, French, Czech, etc.
        - Rarely find *Dampfschifffahrt* in a corpus, but its three morphemes are quite likely
  - Embed **postags, synsets, lexemes, supersenses** (Flekova and Gurevych, 2016), …

Embeddings of other things than words

- Embed **n-grams**
  - That’s the **FastText** approach
  - Bojanowski et al. 2016, Enriching Word Vectors with Subword Information
  - Very popular, available in many languages

- Words are represented as bags of character n-grams (n=3,4,5,6)

  E.g., n=3: where = (>wh , whe, her, ere , re<)
- Embeddings for all n-grams are learned
- Representation for a word is given by average over its n-gram embeddings

- Big advantage:
  - Can embed OOV words, e.g. spelling mistakes: “lenght”, “spellling”
  - Naturally works for morphologically rich languages

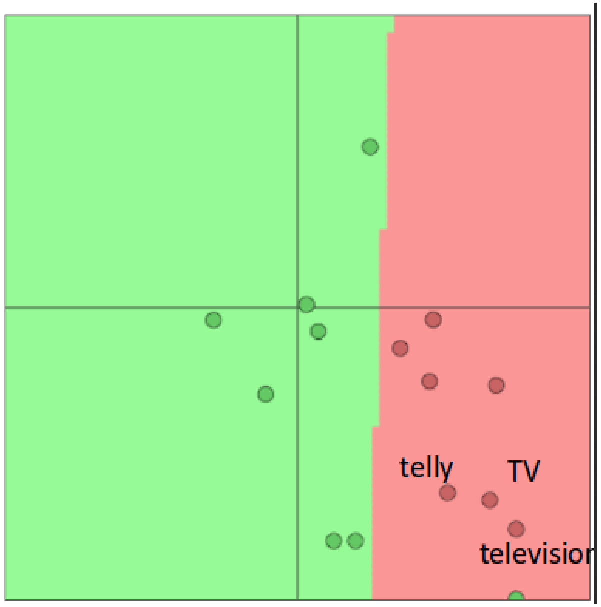
Training word vectors to the task

- Option 1: fixed word representations
  - map word into id and get the vector from the embedding matrix
  - only train the weights of the hidden layers

- Option 2: adjust the word representations to the task
  - word vectors are parameters and are updated in each epoch
  - Example: sentiment classification, train vectors to represent positive/negative polarity for each word

Training word vectors to the task

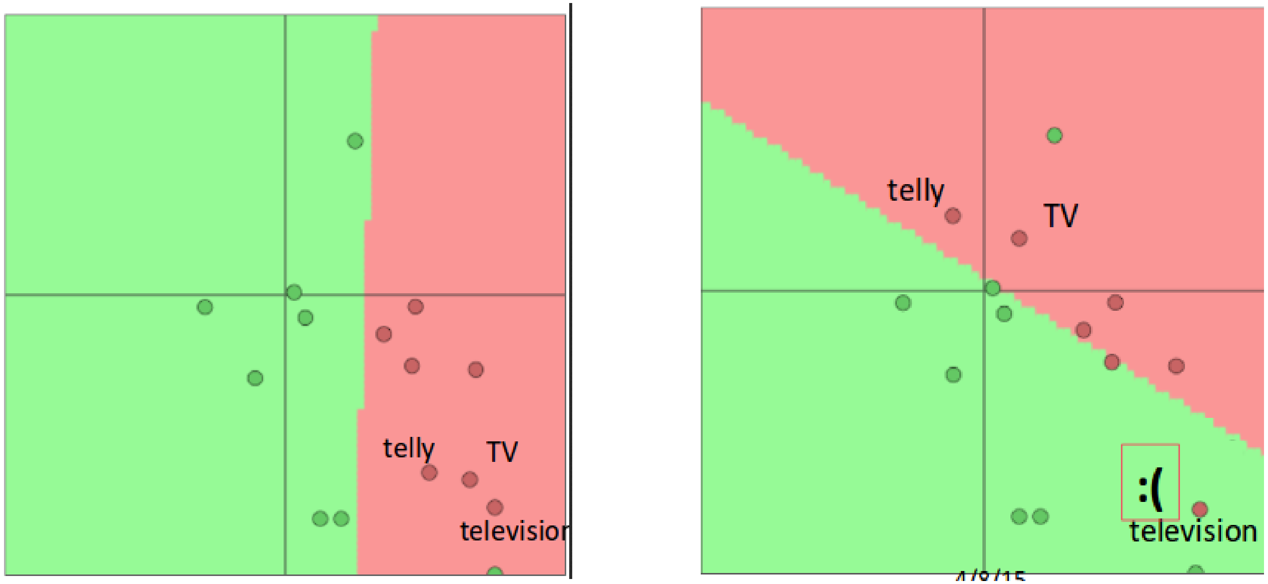
- Problem: Adaptation to the training data
- representations for words that are seen in the training data move in vector space, but words that are not seen remain where they were
- Example by Richard Socher:
  - "TV", "telly" and "television" all indicate negative sentiment in the dataset
  - Due to pre-training, they have similar vectors
  - "TV" and "telly" occur in the training data, "television" in the test data



http://cs224d.stanford.edu/lectures/CS224d-Lecture4.pdf

Training word vectors to the task

- When we train the vectors to the tasks, the words that are seen in the training data move in vector space, but words that are not seen remain where they were.
  - “TV” and “telly” have been updated
  - “television” stayed the same -> synonym information is lost



http://cs224d.stanford.edu/lectures/CS224d-Lecture4.pdf

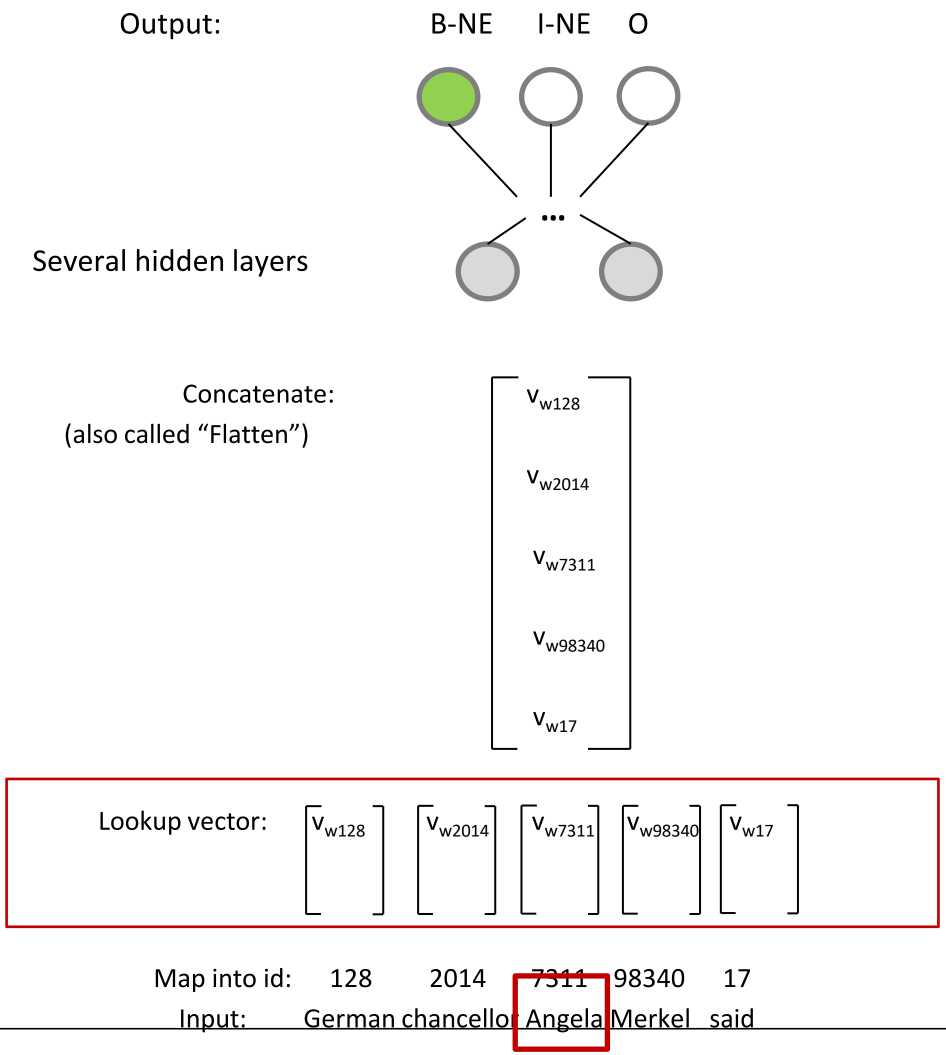
Rule of thumb

- Only train word vectors to the task if you have a large training corpus.
- Even then, it might not be useful (depends on the task).

- ~~Common practice:
  train the vectors only for a few epochs and then keep them fixed~~

If in doubt, don‘t update your embeddings

Using embeddings in a downstream task



Output: B-NE I-NE O

…

Several hidden layers

Concatenate:
(also called “Flatten”)

v_{w128}
v_{w2014}
v_{w7311}
v_{w98340}
v_{w17}

Lookup vector: v_{w128} v_{w2014} v_{w7311} v_{w98340} v_{w17}

Map into id: 128 2014 7311 98340 17

Input: German chancellor Angela Merkel said

The lookup layer

Contextualized word embeddings: ELMo & BERT

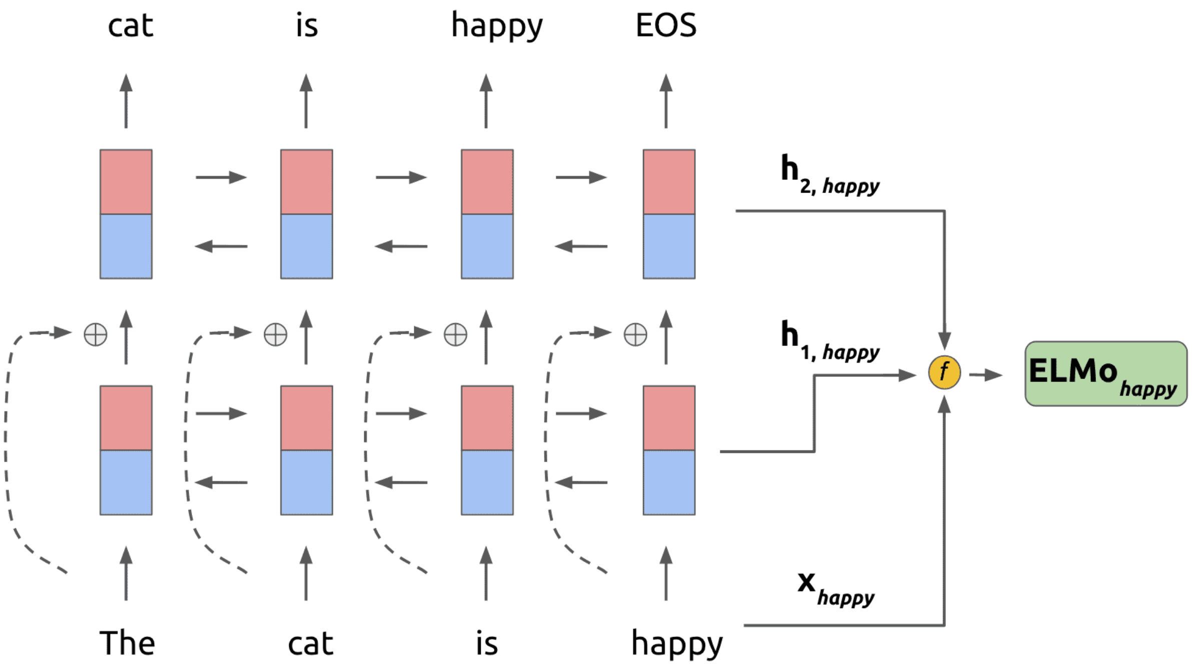
- ELMo and BERT use language models to get **contextualized word representations**: in each context a word has a different embedding
- They are the absolute methods of choice at the moment

- ELMo combines three representations:
  - One on character level
  - Two representations obtained from the two layers in an RNN

- The language model is pre-trained on a large corpus
- For a new task, weights for the three representations are learned to get a task-specific representation
- This task specific representation is concatenated with standard static word embeddings

Contextualized word embeddings: ELMo & BERT

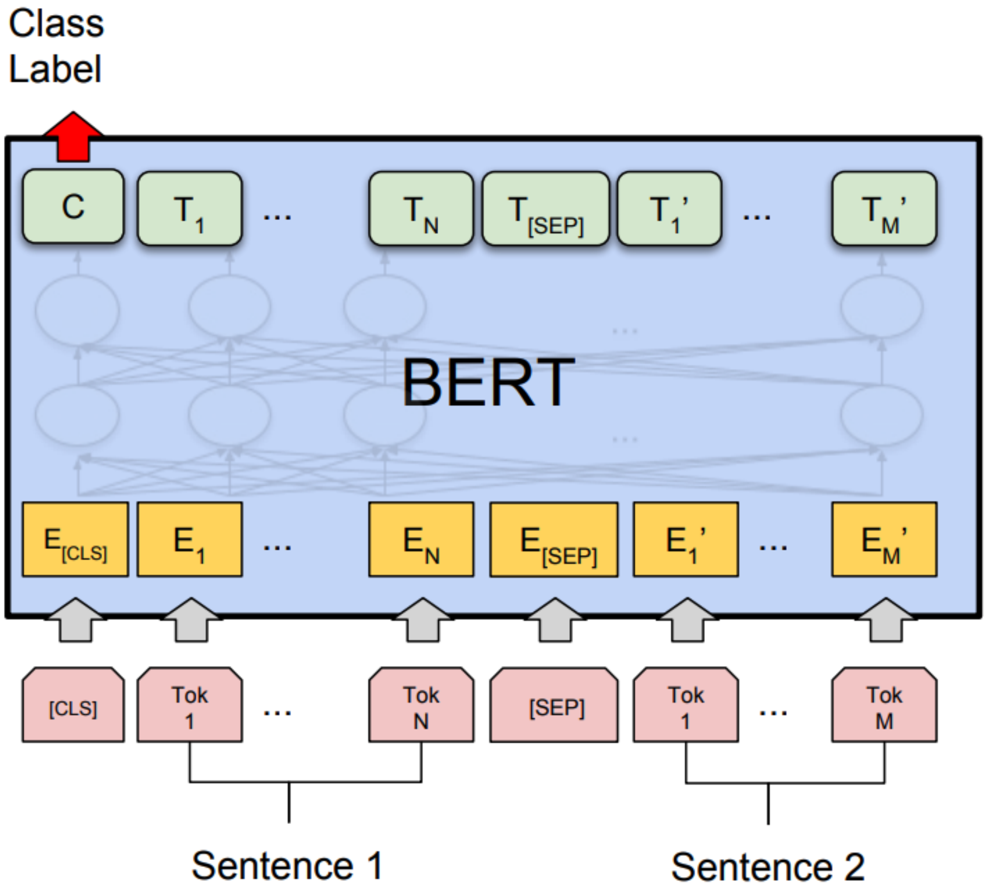
- ELMo visually:



- ELMo (mid-2018) outperformed static word embeddings considerably
- BERT (end-2018) quickly outperformed ELMo
- More on ELMo: https://www.mihaileric.com/posts/deep-contextualized-word-representations-elmo/
- Use BERT for all kinds of tasks: https://github.com/huggingface/pytorch-pretrained-BERT

Contextualized word embeddings: BERT

BERT has changed NLP fundamentally: pre-training & fine-tuning



Summary: Embedding approaches

- What do all the embedding approaches have in common?

  → Represent natural language input with real-valued vectors
- Differences:
  - Unit of representation:
    - characters, morphemes, words, senses, phrases, windows, sentences, documents, …
  - Definition of context for training:
    - CBOW, Skip-gram, Glove, positional, dependency-based, …
- Combining units into larger sequences:
  - More intelligent approaches next week
- Using word representations:
  - as fixed representation for any task (including neural networks but also standard machine learning scenarios e.g. SVMs)
  - as pre-trained initialization in neural networks that gets optimized for the task

Summary: Embedding approaches

- Note that **static word embeddings are becoming extinct** now
- … and replaced by **contextualized embeddings**

Mandatory

- Devlin et al. 2018: **BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding**

References (1)

- Iacobacci, Ignacio, Mohammad Taher Pilehvar, and Roberto Navigli. "SensEmbed: Learning Sense Embeddings for Word and Relational Similarity." *ACL (1)*. 2015.
- Huang, Eric H., et al. "Improving word representations via global context and multiple word prototypes." *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*. Association for Computational Linguistics, 2012.
- Neelakantan, Arvind, et al. "Efficient non-parametric estimation of multiple embeddings per word in vector space." *arXiv preprint arXiv:1504.06654* (2015).
- Luong, Thang, Hieu Pham, and Christopher D. Manning. "Bilingual word representations with monolingual quality in mind." *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*. 2015.
- Hermann, Karl Moritz, and Phil Blunsom. "Multilingual distributed representations without word alignment." *arXiv preprint arXiv:1312.6173* (2013).
- Vulic, Ivan, and Marie-Francine Moens. "Bilingual word embeddings from non-parallel document-aligned data applied to bilingual lexicon induction." *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL 2015)*. ACL, 2015.

References (2)

- Upadhyay, Shyam, et al. "Cross-lingual models of word embeddings: An empirical comparison." *arXiv preprint arXiv:1604.00425* (2016).
- Klementiev, Alexandre, Ivan Titov, and Binod Bhattarai. "Inducing crosslingual distributed representations of words." (2012).
- Upadhyay, Shyam, et al. "Cross-lingual models of word embeddings: An empirical comparison." *arXiv preprint arXiv:1604.00425* (2016).
- Bengio, Yoshua, and Greg Corrado. "Bilbowa: Fast bilingual distributed representations without word alignments." (2015).
- Ling et al. 2015: Two/Too Simple Adaptations of Word2Vec for Syntax Problems
- Levy and Goldberg, 2014: Dependency-Based Word Embeddings
- Komninos, Alexandros, and Suresh Manandhar. "Dependency based embeddings for sentence classification tasks." *Proceedings of NAACL-HLT*. 2016.