Letter-based Composition of German Compounds

Corina Dima and Jianqiang Ma
University of Tübingen, SFB 833/**A3**

Outline

- How to Represent Compounds?
- Distributional Word Representations
- Compositional Representations for German Compounds
  - lemma-based composition
  - letter ngram-based composition

How to Represent Compounds?

- **Task**: Study the semantics of German compounds using computational approaches

- **Problem**: Find good representations for compounds that:
  - capture the meaning of compounds
  - can represent **any compound** that **was or will be coined** by the speakers of the language

Productivity and Low Frequency of Compounds

- Baroni et al. (2002) analyzed the 28 million words German APA news corpus and discovered that compounds account for **47%** of the **word types**, but only **7%** of the overall **token count** are compounds

- **83%** of compounds have a corpus frequency of 5 or lower

- **plus: newly coined** compounds will **never** appear in a corpus

Two Problems

1. How do we represent the meaning of (frequent enough) words?

2. How do represent the meaning of infrequent/newly coined words?

Solution for #1: Distributional Semantics

wampimuk

He filled the *wampimuk*, passed it around and we all drunk some

We found a little, hairy *wampimuk* sleeping behind the tree

- Example from McDonald & Ramscar (2001), photos from Google Image

Distribution Semantics: Intuition

"You shall know a word by the company it keeps"

-- Firth (1957)

John R. Firth

- Words that occur in similar **contexts** are semantically similar
- We can approximate the meaning of a word using the **contexts** it appears in

Distributional Semantics: Co-occurrence count

context

Don't know. T: OK, let's try this. If a car was driving along east, which way would
including all flights, hotels, and rental cars , all in conversational English over the
including the booking of each flight, hotel, and car reservation. Because the number of legs

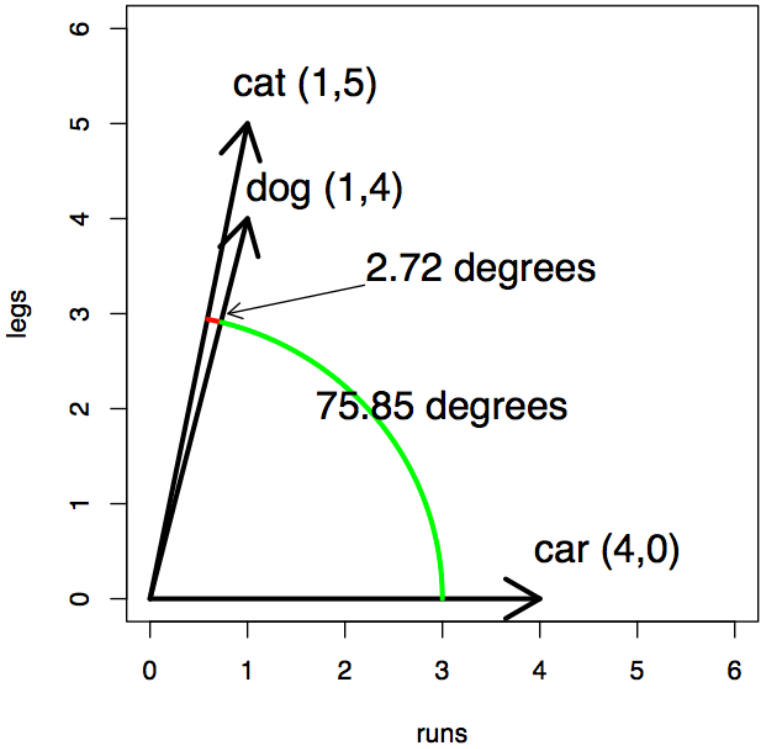
*screenshot from Sketch Engine, ACL Anthology Reference Corpus

	car	bus	plane
ticket	0	4	2
drive	4	3	0
fly	0	0	5
passenger	2	4	3

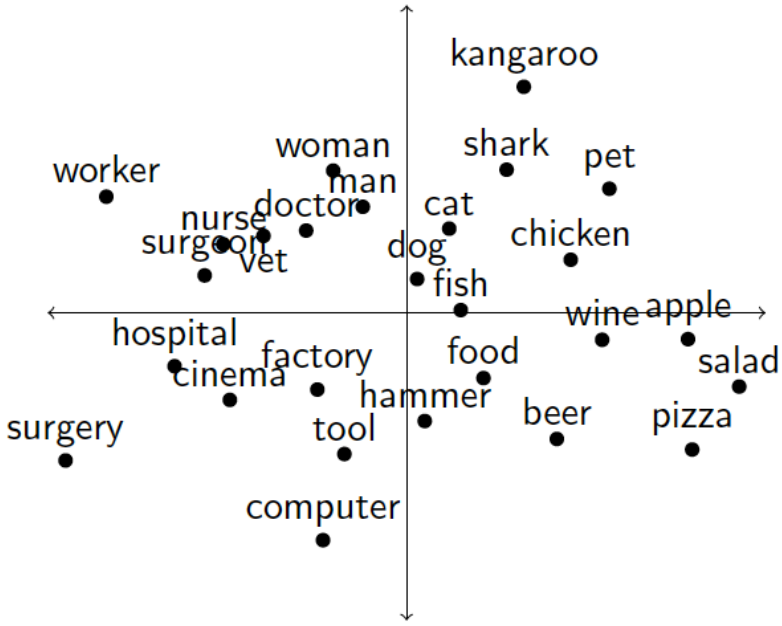
car	[0,4,0,2]
bus	[4,3,0,4]
plane	[2,0,5,3]

Distributional Semantics: Similarity

- Semantic similarity as vector space similarity



Baroni & Boleda. *Distributional Semantics*. http://bit.ly/1Kx0Qer



O S´eaghdha. 2011. *Distributional Approaches to Semantic Analysis*. http://bit.ly/1MKKTFa

2 Problems

1. How do we represent the meaning of (frequent enough) words? ✔

2. How do represent the meaning of infrequent/newly coined words?

Distributional Semantics: Details in a Nutshell

- **Context** matters
  - Words in a window of size 2, 5, …100
  - Syntactic context
  - Documents
- Dimensionality reduction via PCA, etc.
  - Offset sparseness in ~10^6 of dimensions
  - Improve computational efficiency
- Similarity measures
  - Cosine similarity
- Re-weight counts
  - PMI, TF/IDF…

	car	bus	plane
ticket	0	4	2
drive	4	3	0
fly	0	0	5
passenger	2	4	3
…	…	…	…
cat	0	0	0
forest	0	0	0
hay	0	0	0
stone	0	0	0

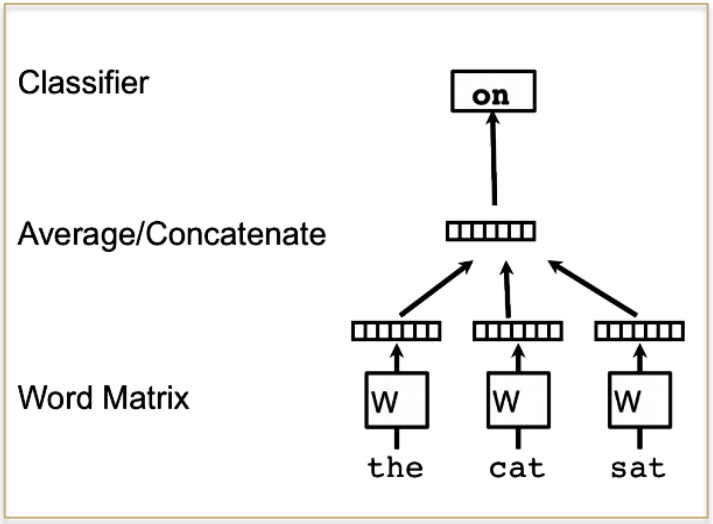
vocabulary

Distributional Semantics: Predict!

- **Word embedding** (Bengio et. al, 2003; Collobert & Weston, 2008)
  n-dimensional, real-valued vector that are learned by *predicting* other words.

word2vec (Mikolov et al. 2013): Predict!

- Train vectors to either:
  - Predict a word given its bag-of-words contexts (**CBOW**)
  - Predict a context word from the center word (Skip-gram)
- Update word vectors until they can do this prediction well



CBOW

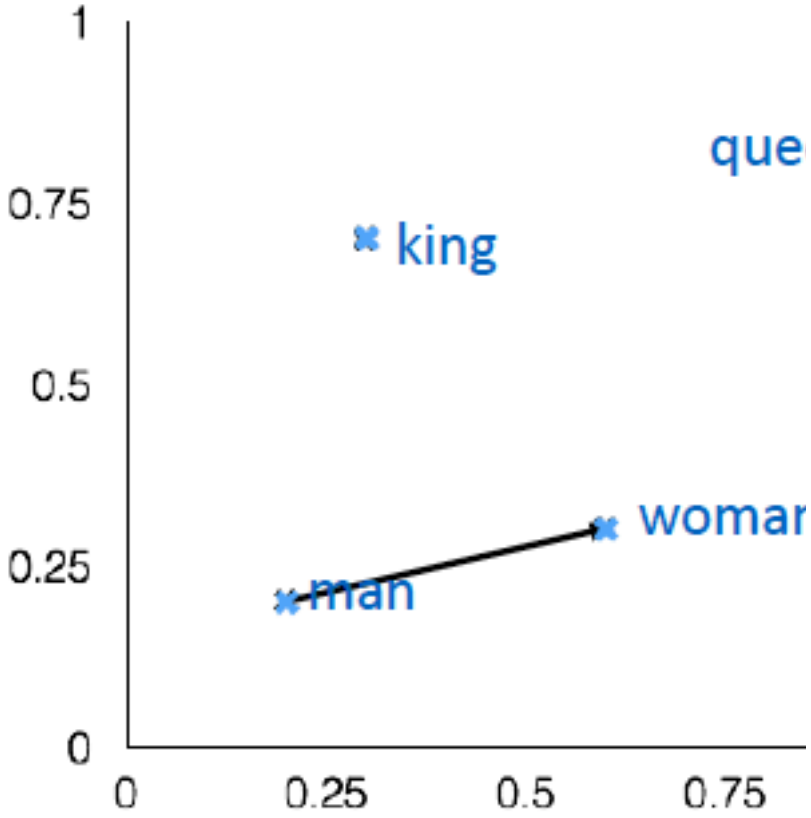
… *the cat sat on the mat* …

figure from: http://bit.ly/1NOOm4k

Word Embedding: “king – man + woman = queen”

an:woman :: king:?

ng	[0.30 0.70]
an	[0.20 0.20]
oman	[0.60 0.30]
ueen	[0.70 0.80]



Chris Manning. 2015. *Compositional Deep Learning*. http://stanford.io/1WkJDub

Word Embedding: Directions in Vector Space

Word Analogies: relations as linear operation of vectors

Relationship	Example 1	Example 2	Example 3
France - Paris	Italy: Rome	Japan: Tokyo	Florida: Tallahassee
big - bigger	small: larger	cold: colder	quick: quicker
Miami - Florida	Baltimore: Maryland	Dallas: Texas	Kona: Hawaii
Einstein - scientist	Messi: midfielder	Mozart: violinist	Picasso: painter
Sarkozy - France	Berlusconi: Italy	Merkel: Germany	Koizumi: Japan
copper - Cu	zinc: Zn	gold: Au	uranium: plutonium
Berlusconi - Silvio	Sarkozy: Nicolas	Putin: Medvedev	Obama: Barack
Microsoft - Windows	Google: Android	IBM: Linux	Apple: iPhone
Microsoft - Ballmer	Google: Yahoo	IBM: McNealy	Apple: Jobs
Japan - sushi	Germany: bratwurst	France: tapas	USA: pizza

Distributional Semantics: Various Approaches

- **Count!** : GloVe, Hellinger-PCA (lebret) …
- **Predict!** : word2vec, Collobert & Weston (CW) …

coffee

cw50	glove300_42b	lebret200	word2vec300_100b	cw50-glove300_42b-lebret200
coffee shop	tea	sugar	tea	tea
tea	drink	cotton	chocolate	sugar
cooking	cafe	wool	cocoa	beer
cider	chocolate	beef	soda	wine
wine	breakfast	meat	beer	milk

arm

cw50	glove300_42b	lebret200	word2vec300_100b	cw50-glove300_42b-lebret200
wing	arms	chain	leg	arms
body	leg	sphere	shoulder	body
weapon	hand	body	wrist	leg
suit	shoulder	raft	elbow	wing
head	neck	stream	arms	side

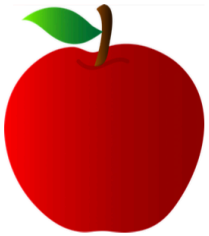
Code and Resources

- Predict! models (learning word embeddings)
  - Word2vec: code.google.com/p/word2vec
    - with downloadable English word vectors
  - RNNLM: www.fit.vutbr.cz/~imikolov/rnnlm

- Count! Models (traditional distributional semantics)
  - COMPOSES: http://clic.cimec.unitn.it/composes
  - GloVe: http://nlp.stanford.edu/projects/glove/
    - with downloadable English word vectors
    - German word vectors: contact Corina Dima @A3 Project

Solution A for #2: Lemma-based Compositional Representations for German Compounds

Compound	Lemma-based representation
Apfelbaum	[apfel][baum]
Holzlöffel	[holz][löffel]
Abendkleid	[abend][kleid]
Chefkoch	[chef][koch]
Mandelöl	[mandel][öl]
Orangensaft	[orange][saft]
Gästebad	[gast][bad]
Hochzeitsbild	[hochzeit][bild]
Ameisenbären	[ameise][bär]


Apfel

+


Baum

→


Apfelbaum

0.3	0.1	0.7	1.3	0.2	u	0.5	0.9	0.1	0.4	1.2	v	0.2	1.0	0.6	0.7	1.1	w

$$f\left(\begin{array}{|c|c|c|c|c|}\hline 0.3 & 0.1 & 0.7 & 1.3 & 0.2 \\ \hline\end{array}\, u\,,\ \begin{array}{|c|c|c|c|c|}\hline 0.5 & 0.9 & 0.1 & 0.4 & 1.2 \\ \hline\end{array}\, v\right) = \begin{array}{|c|c|c|c|c|}\hline ?? & ?? & ?? & ?? & ?? \\ \hline\end{array}\, p$$

What f makes p most similar to w?

Dataset

34497 compounds from GermaNet 9.0 German compounds list; frequency filtered: modifier, head and compound with min. frequency 500 in the support corpus.

Word Representations

Trained 50, 100, 200 and 300 word representations using GloVe (Pennington et. al, 2014), a 10B token raw-text corpus extracted from the DECOW14AX corpus (Schäfer, 2015) and 1M words vocabulary.

12 composition functions

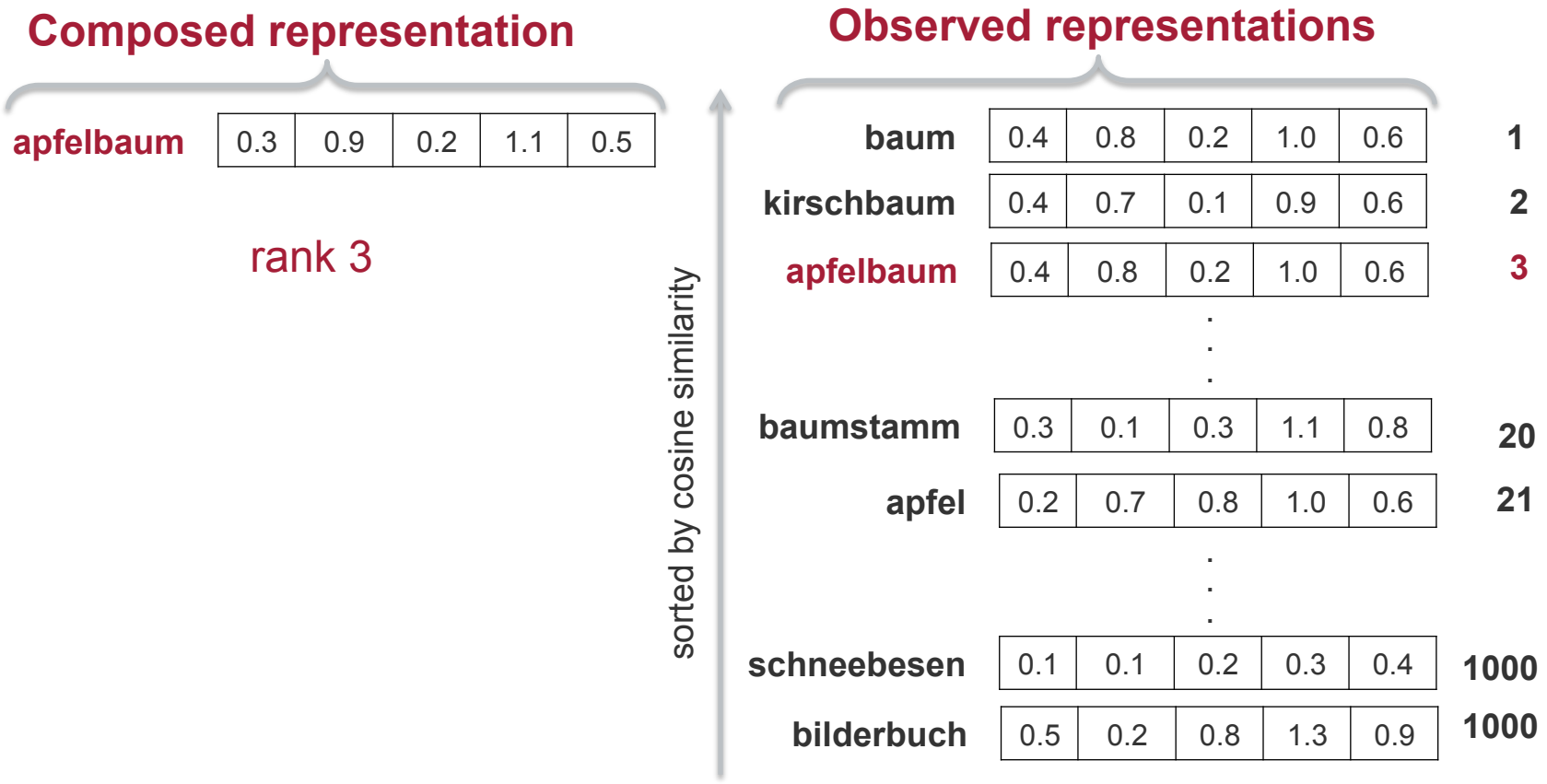
No	Formula	Name
1.	$\mathcal{p} = v$	head
2.	$\mathcal{p} = u$	modifier
3.	$\mathcal{p} = u \odot v$	component-wise multiplication
4.	$\mathcal{p} = (u \cdot u)v + (\lambda\text{-}1)(u \cdot v)u$	dilation
5.	$\mathcal{p} = 0.5u + 0.5v$	addition
6.	$\mathcal{p} = \lambda u + \beta v$	weighted addition
7.	$\mathcal{p} = \mathcal{U}v$	lexical function
8.	$\mathcal{p} = \mathcal{M}_1 u + \mathcal{M}_2 v$	full additive
9.	$\mathcal{p} = g(\mathcal{W}[u;v])$	matrix
10.	$\mathcal{p} = g(\mathcal{W}[\mathcal{V}u; \mathcal{U}v])$	full lexical
11.	$\mathcal{p} = u \odot u' + v \odot v''$	additive mask
12.	$\mathcal{p} = g(\mathcal{W}[u \odot u'; v \odot v''])$	**W mask**

$u, v, p \in \mathbb{R}^n$; $\lambda, \beta \in \mathbb{R}$; $\mathcal{U}, \mathcal{V}, \mathcal{M}_1, \mathcal{M}_2 \in \mathbb{R}^{n \times n}$; $\mathcal{W} \in \mathbb{R}^{n \times 2n}$; $g = tanh$

How does the lemma-based composition work?

apfel	0.2	0.5	-0.1	0.3	0.9
			+		
baum	0.1	0.4	0.3	0.8	-0.4
			=		
apfelbaum	0.3	0.9	0.2	1.1	0.5

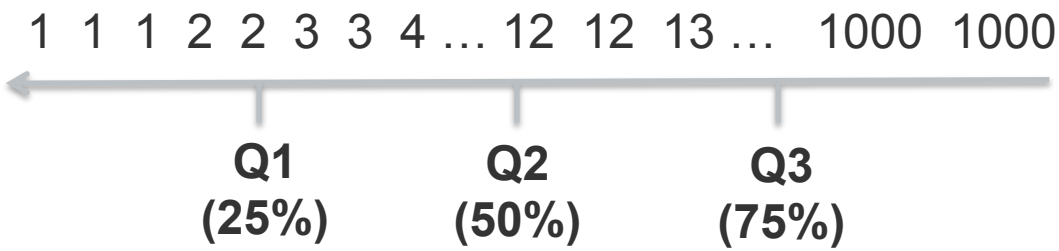
Evaluating Composition Models



- the list of observed representations includes all the compounds as well as the modifiers and heads – 41732 words in total

Evaluating Composition Models (2)

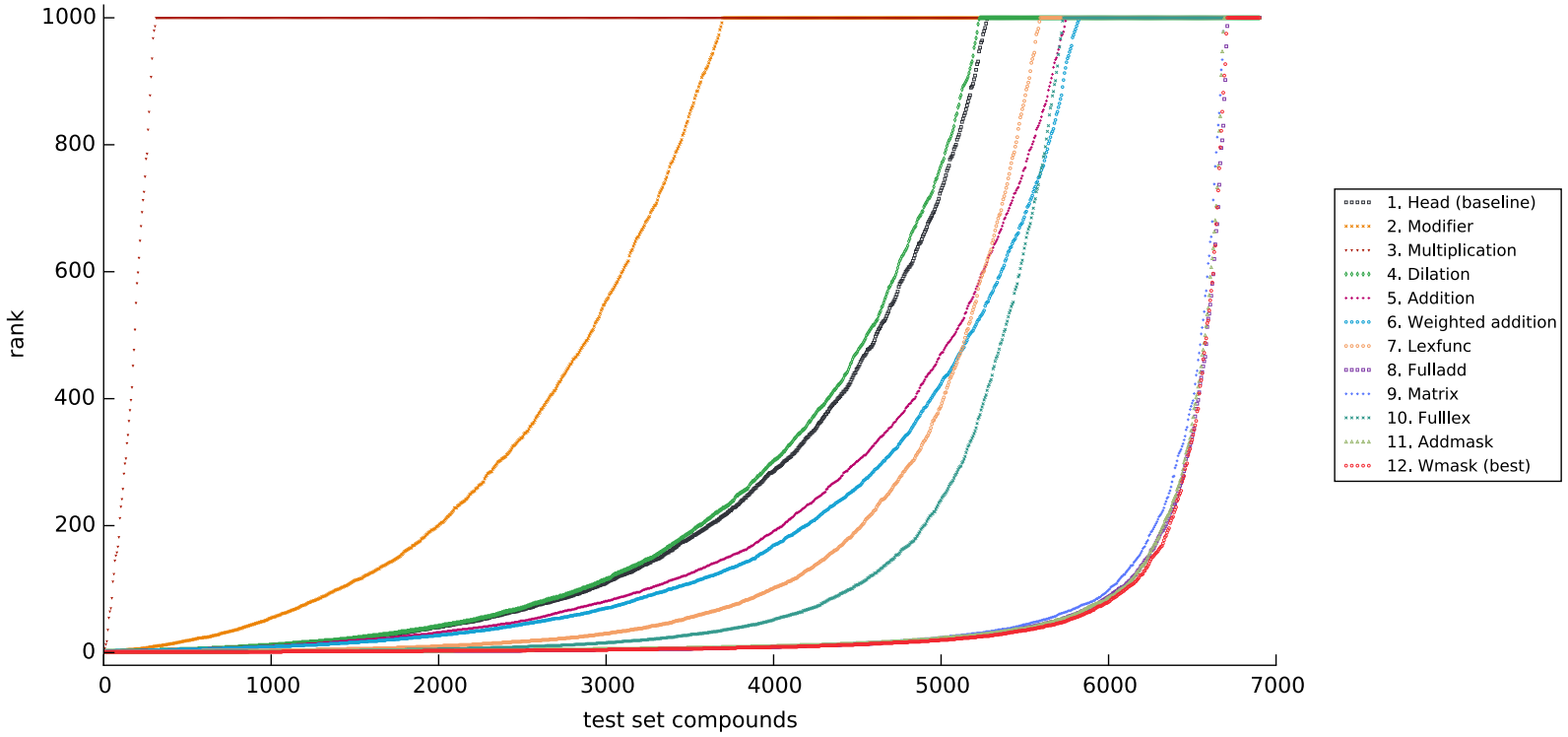
telefonkabel 1
sehstörung 1
exportanteil 1
schlossturm 1
…
hundekuchen 12
vorplatz 12
arbeitsposition 12
sitzheizung 12
…
maulwurf 1000
milchmädchen 1000
frauenschuh 1000
rosenblatt 1000



Results for Lemma-based Composition

		Q1	Q2	Q3
5	$p = 0.5u + 0.5v$ vector addition[1]	24	120	553
6	$p = \lambda u + \beta v$ weighted vector addition[1]	20	105	503
7	$p = \mathcal{U}v$ lexical function[2,3]	7	51	526.5
8	$p = \mathcal{M}_1 u + \mathcal{M}_2 v$ full additive[4,3]	**2**	**6**	27
9	$p = g(\mathcal{W}[u;v])$ matrix[5]	**2**	7	29
10	$p = g(\mathcal{W}[\mathcal{V}u; \mathcal{U}v])$ full lexical[6,3]	4	26	334
NEW! 11	$p = u \odot u' + v \odot v''$ additive mask	3	7	27
NEW! 12	$p = g(\mathcal{W}[u \odot u'; v \odot v''])$ Wmask	**2**	**6**	**24**

Results



- evaluation: lower **rank** is better;
- ideal model would have a flat line on rank 1

Problems with Lemma-based Composition

- real compounds don't come already split
- lemma-based composition models assign the same representation for different compounds (singular and plural compounds have the same representations)

Compound	Lemma-based representation
Orange**n**saft	[orange][saft]
G**ä**st**e**bad	[gast][bad]
Hochzeit**s**bild	[hochzeit][bild]
Ameise**n**bär**en**	[ameise][bär]
Ameise**n**bär	[ameise][bär]

Solution B for #2: Letter-based Compositional Representations for German Compounds

- Problem: real compounds don't come already split
  Idea: Get compound representation *without* splitting.

- Problem: lemma-based composition models assign the same representation for different compounds (singular and plural compounds have the same representations)
  Idea: Keep the form of a compound as it is, including linking element etc.

Letter-based Compositional Representations for German Compounds

Main Idea: Associate word form with meaning i.e. morphology to semantics

- Intuition: Certain letters in words convey syntactic and semantic information
  - Affix: suffix such as '-ung', '-er' or prefix 'ge-'
  - Word/compound as a whole
  - Beyond morphemes/lemmas: any letter combinations: can you think of some examples?

- How: Enumerate candidates and let the statistical model decide which letters are important.

Letter *N-Gram* Representation of Compound Form

Word: apfelbaum

1-grams: *#, a, p, f, e, l, b, a, u, m, #*
2-grams: *#a, ap, pf, fe, el, lb, ba, au, um, m#*
3-grams: *#apf, apf, pfe, fel, elb, lba, bau, aum, um#*
4-grams: *#apf, apfe, pfel, felb, elba, lbau, baum, aum#*
5-grams: *#apfe, apfel, pfelb, felba, elbau, lbaum, baum#*

***n*-gram** combinations: 1+2 gram, 2+3 gram, 3+4 gram, 4+5gram;
1+2+3gram, 2+3+4gram, 3+4+5gram

- All letter combinations up to length *n*, not really morphemes.

How to Build One-Hot Representations

- step 1: build vocabularies for different n-gram lengths

1-grams	=	34
2-grams	=	~700
3-grams	=	~7000
4-grams	=	~30000
5-grams	=	~62000

- step 2: check which of the ngrams in the vocabulary occur in a particular word

- step 3: build a 0/1 vector as a representation for our word

Example: How to Build One-Hot Representations

word: apfelbaum

1-grams: *a, p, f, e, l, b, a, u, m*

Which letters in the alphabet occur in 'apfelbaum' ?

a= 1, b=1, c=0, d=0, e=1, f=1, g=0,…, k=0, l=1, m=1,
…, n=0, o=0, p=1, q=0…s=0, u=1, v=0, …z =0

a b e f l m p u

[1, 1, 0, 0, 1, 1, 0, …0, 1, 1, 0, …0, 1, 0…1,.., 0]

Map occurrences to "1"s in corresponding positions of the vector

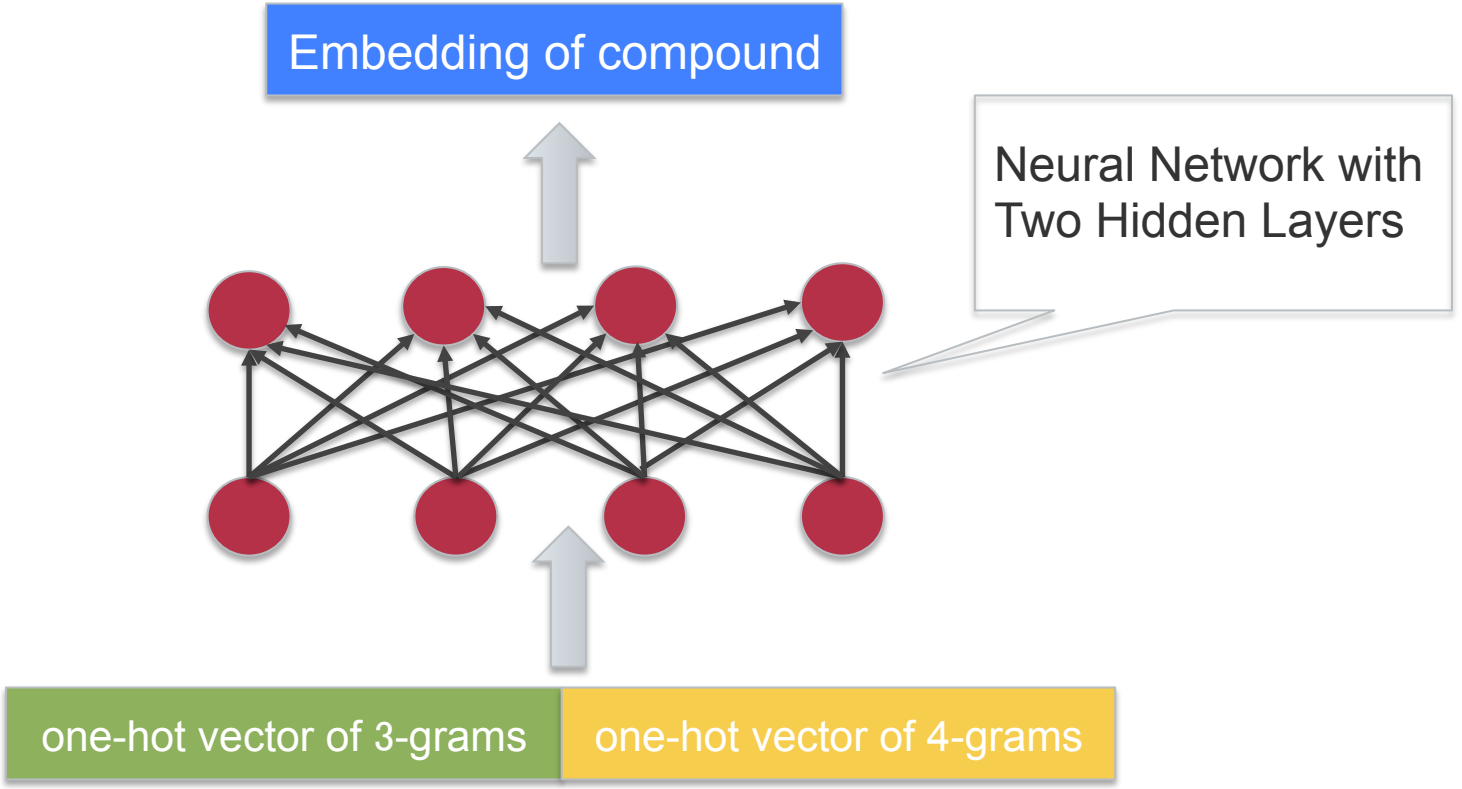
Letter N-Gram One-HOT Representations

Recap: One-hot vector for *1-gram*

- A vector of length m, m= alphabet size.
- The value of at each position of the vector represents the occurrence of the corresponding letter in the alphabet.
- Value = 1 means occurrence (can be multiple times); value=0 means absence.

- One-hot vectors can represent 2-gram, 3-gram, etc. in the same manner
- Our letter n-gram representation: *concatenation* of one-hot vectors of several n-gram, e.g. 2-gram and 3-gram

Compose Letter N-gram Representation to Get Semantic Representation of Compound



Letter−Ngram based Composition Model Details

- **Neural network model**
  - 2-hidden layers, each has 1000 hidden units
  - Relu activation for hidden layers
  - 2 dropout layers with 10% dropout rate

- **Loss function**: *mean squared error* or *cosine distance*

- **Learning algorithm**: Mini-batch stochastic gradient descent, with Adagrad

Experiment Setup

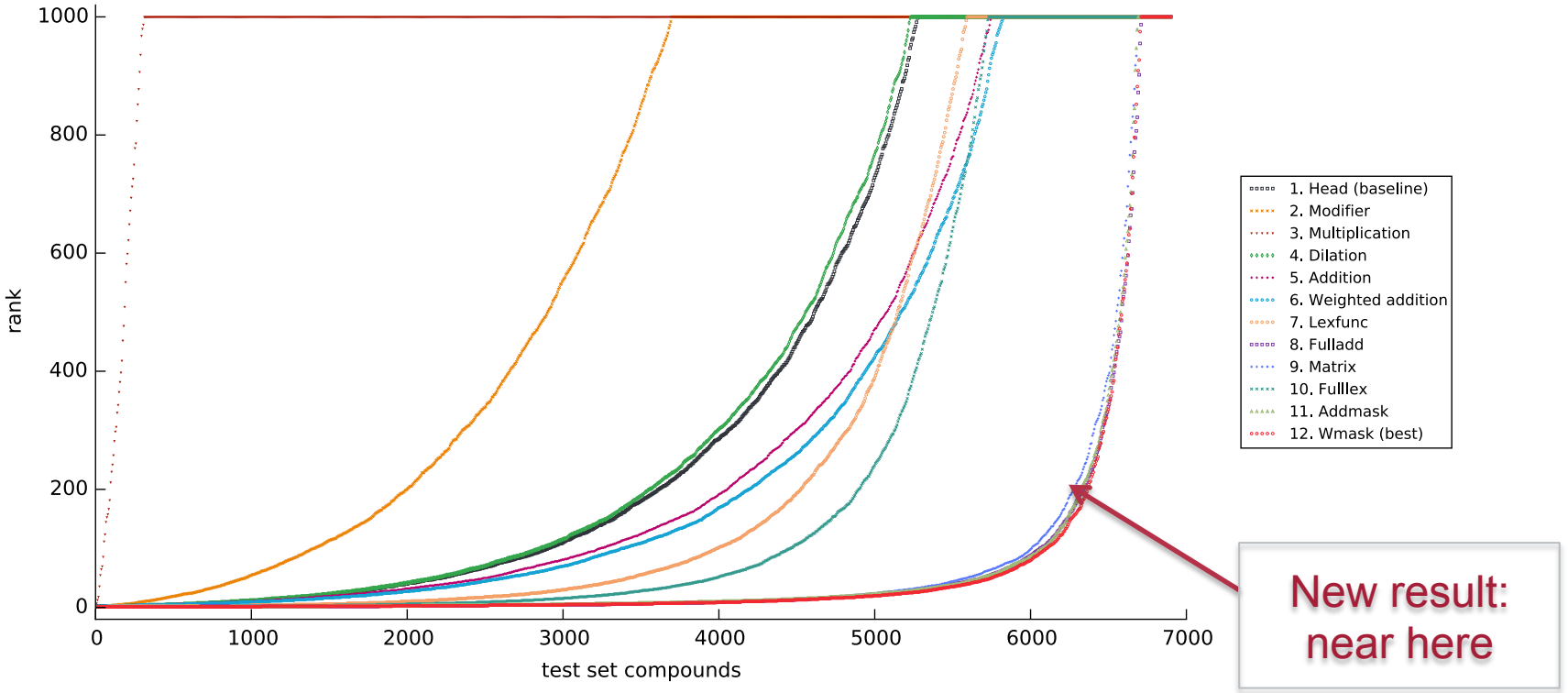
- Same dataset as in lemma-based composition experiments
- Also same evaluation metric: rank w.r.t. the full lexicon

- Differences in input:
  - Letter ngrams of a compound as input
  - No splitting of compound
  - No knowledge about the head/modifier or their embeddings

- The model learns to map the form of a compound (i.e. letter ngrams) to its meaning (word embedding)

Results: Comparison with Lemma-Based Compsoition

- On par with top-4 models for lemma based composition



Results: Rank per 1/4 Quantile (lower is better)

Ngrams used	Q1	Q2	Q3
1-gram	> 1000	> 1000	> 1000
2-gram	48	352	> 1000
3-gram	6	30	264
4-gram	**3**	14	94
5-gram	**3**	13	88
1- & 2-gram	70	512	1000
2- & 3-gram	6	37	304
3- & 4-gram	**3**	14	96
4- & 5-gram	**3**	**12**	**72**
(L) vector addition	**24**	**120**	**553**
(L) wmask	**2**	**6**	**24**

Discussion of Letter-Based Composition

Mapping word form to meaning representation

- Naïve morphology: use n-grams up to 5 letters to represent words.
- Surprisingly good results given that it uses no semantic information of constituents of a compound.
- It works even if *no* compound is used in training (detailed results omitted).
  - Train with lexicon – all compounds
  - Train with lexicon – all compounds – all words that are heads or modifiers of compounds.

Discussion of Letter-Based Composition (2)

As general NLP representation & composition approach?

- Letter ngrams are robust for rare words and non-canonical languages
- Compound is above word-level, similar to phrase. Good results on compounds even if no compound is used in training.
- Neural network model as a general way of compositionality.
- Future work: try the model on other language applications.

Wrap Up

- Distributional semantics offers a way to represent the meaning of a word

- Representations for compounds can be composed starting from:
  - lemmas of the constituent words
  - letter ngrams of the whole compound

- Advantages of the letter-based composition:
  - no need for compound splitter
  - different representations for singular and plural form of a compound
  - can be extended to phrase and sentence representations

Questions ?

SFB 833/**A3**

@Corina Dima
corina.dima@uni-tuebingen.de

@Jianqiang Ma
jma@sfs.uni-tuebingen.de