Splitting Compounds By Semantic Analogy

Joachim Daiber, Lautaro Quiroz, Roger Wechsler and Stella Frank

Institute for Logic, Language and Computation
University of Amsterdam

Lautaro Quiroz
In Master of AI, UvA

Roger Wechsler
In Master of AI, UvA

Dr. Stella Frank
ILLC, UvA

Introduction

Compound words...

- ... make life hard for standard NLP applications, incl. MT
- ... are often modeled with shallow information (e.g. Moses frequency-based splitter)

Question: Can we use distributional semantics to do deeper processing of compounds in a simple way?

Splitting compounds for SMT

- Koehn and Knight (2003) showed PBMT systems can better deal with compounds if they are split into their meaningful parts
- Difficulty: many possible splits, we need to choose the correct ones

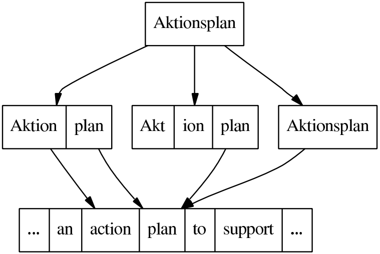


Figure: Compound splitting example from Koehn and Knight (2003).

Compounds and the semantic vector space

Semantic vector space

- Word embeddings saw surge of successful applications recently
- Basic idea: "You shall know a word by the company it keeps"
  - Words are mapped to vectors of real numbers in low dimensional space
  - These vectors are estimated on large amounts of text data using a neural network

Compounds and the semantic vector space

Semantic vector space

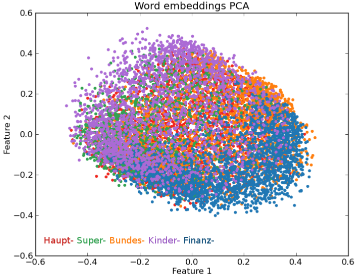
- Mikolov et al. (2013) showed that word embeddings capture some linguistic phenomena:
  - *king* is to *man* what *queen* is to *woman*
    $v(\text{king}) - v(\text{man}) + v(\text{woman}) \approx v(\text{queen})$
  - *cars* is to *car* what *dogs* is to *dog*
    $v(\text{cars}) - v(\text{car}) + v(\text{dog}) \approx v(\text{dogs})$

Compounds and the semantic vector space

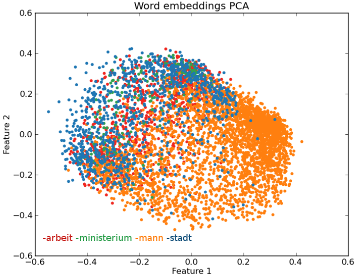
Morphology induction from word embeddings

- Soricut and Och (2015) exploit these regularities to induce morphology from word embeddings
- Method:
  - Extract prefix and suffix replacement rules from the vocabulary
  - Keep 1000 examples of each rule
  - Judge how well each pair explains the other pairs: *cars* is to *car* what *dogs* is to *dog*?
  - Find most representative examples for each rule

Compounds and the semantic vector space



(a) Compounds with same modifier.



(b) Compounds with the same head.

The analogy test

- We model compounds based on their modifiers
- Potential compound splits are judged by how similar they are to a set of prototypical compounds for each modifier

Analogy test: *Mauszeiger* is to *Zeiger* what *Mausklick* is to *Klick*?

(mouse pointer) (pointer) (mouse click) (click)

Extracting potential compound splits

For all words in the vocabulary:

- Extract all possible string prefixes ≥ 4:
  Bundespräsident $\rightarrow$ *Bund, Bunde, Bundes, ...*
- Judge each Modifier+Compound pair by how well it explains others

Judging potential compound splits

All potential compounds with prefix *Maus*

Maus|kostüm
Maus|zeiger
Maus|stämme
Maus|klick
Maus|hirn
Maus|tasten
Maus|ersatz
Maus|mutanten
Maus|knopf
Maus|steuerung
Maus|bewegung
Maus|gene
Maus|klicks
Maus|hirns
Maus|zeiger
Maus|hirnen
Maus|bedienung

...
(up to 500)

Judging potential compound splits

All potential compounds $\times$ All potential compounds

Maus|kostüm	Maus|kostüm
Maus|zeiger	Maus|zeiger
Maus|stämme	Maus|stämme
Maus|klick	Maus|klick
Maus|hirn	Maus|hirn
Maus|tasten	Maus|tasten
Maus|ersatz	Maus|ersatz
Maus|mutanten	Maus|mutanten
Maus|knopf	Maus|knopf
Maus|steuerung	Maus|steuerung
Maus|bewegung	Maus|bewegung
Maus|gene	Maus|gene
Maus|klicks	Maus|klicks
Maus|hirns	Maus|hirns
Maus|zeiger	Maus|zeiger
Maus|hirnen	Maus|hirnen
Maus|bedienung	Maus|bedienung
...	...
(up to 500)	(up to 500)

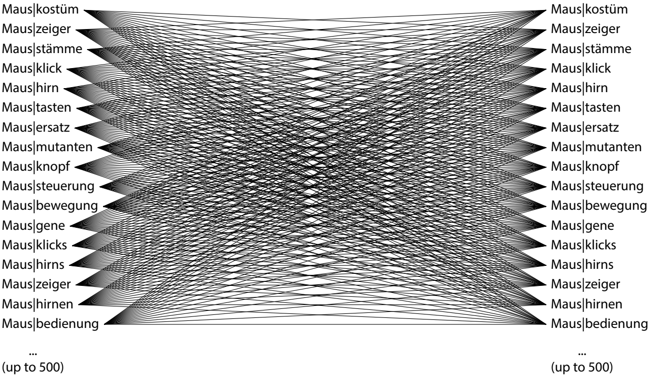
Judging potential compound splits

All potential compounds $\times$ All potential compounds

Maus|kostüm	Maus|kostüm
Maus|zeiger	Maus|zeiger
Maus|stämme	Maus|stämme
Maus|klick	Maus|klick
Maus|hirn	Maus|hirn
Maus|tasten	Maus|tasten
Maus|ersatz	Maus|ersatz
Maus|mutanten	Maus|mutanten
Maus|knopf	Maus|knopf
Maus|steuerung	Maus|steuerung
Maus|bewegung	Maus|bewegung
Maus|gene	Maus|gene
Maus|klicks	Maus|klicks
Maus|hirns	Maus|hirns
Maus|zeiger	Maus|zeiger
Maus|hirnen	Maus|hirnen
Maus|bedienung	Maus|bedienung
...	...
(up to 500)	(up to 500)

Judging potential compound splits

All potential compounds $\times$ All potential compounds



Judging potential compound splits

Maus|zeiger	Maus|zeiger
Maus|stämme	Maus|stämme
Maus|klick	Maus|klick
Maus|hirn	Maus|hirn

Perform analogy test: *Mauszeiger* is to *Zeiger* what *Mausklick* is to *Klick*?
(mouse pointer) (pointer) (mouse click) (click)

Computational considerations

- **Analogy test is expensive!**
- True and predicted vectors:
  - $v_{\text{Mausklick}}$
  - $\hat{v}_{\text{Mausklick}} = \text{Mauszeiger} - \text{Zeiger} + \text{Klick}$
- Two evaluation functions: $\textsc{Rank}$ and $\textsc{Cosine}$

Computational considerations

- Exact but slow implementation:

$$\textsc{Rank}(v_{cmpd}, \hat{v}_{cmpd}) = \textsc{Rank of } v_{cmpd} \textsc{ in } \operatorname*{arg\,sort}_{w \in V} \left[\textsc{Cosine}\left(v_w, \hat{v}_{cmpd}\right) \right]$$

- Approximate but fast implementation:
  - Approximate k-nearest neighbor search
  - We use the Spotify Annoy library (C++) to perform the search

Prototypes

Compounds that are good examples of a compound modifier.

- These are best at explaining other similar modifier+compound pairs
- We call this set the modifier's ***prototypes***

Extracting prototypes

Maus|kostüm
Maus|zeiger
Maus|stämme
Maus|klick
Maus|hirn
Maus|tasten
Maus|ersatz
Maus|mutanten
Maus|knopf
Maus|steuerung
Maus|bewegung
Maus|gene
Maus|klicks
Maus|hirns
Maus|zeiger
Maus|hirnen
Maus|bedienung
...
(up to 500)

Maus|kostüm
Maus|zeiger
Maus|stämme
Maus|klick
Maus|hirn
Maus|tasten
Maus|ersatz
Maus|mutanten
Maus|knopf
Maus|steuerung
Maus|bewegung
Maus|gene
Maus|klicks
Maus|hirns
Maus|zeiger
Maus|hirnen
Maus|bedienung
...
(up to 500)

Extracting prototypes

Maus|kostüm	Maus|kostüm
Maus|zeiger	Maus|zeiger
Maus|stämme	Maus|stämme
Maus|klick	Maus|klick
Maus|hirn	Maus|hirn
Maus|tasten	Maus|tasten
Maus|ersatz	Maus|ersatz
Maus|mutanten	Maus|mutanten
Maus|knopf	Maus|knopf
Maus|steuerung	Maus|steuerung
Maus|bewegung	Maus|bewegung
Maus|gene	Maus|gene
Maus|klicks	Maus|klicks
Maus|hirns	Maus|hirns
Maus|zeiger	Maus|zeiger
Maus|hirnen	Maus|hirnen
Maus|bedienung	Maus|bedienung
...	...
(up to 500)	(up to 500)

Extracting prototypes

Maus|kostüm	Maus|kostüm
Maus|zeiger	Maus|zeiger
Maus|stämme	Maus|stämme
Maus|klick	Maus|klick
Maus|hirn	Maus|hirn
Maus|tasten	Maus|tasten
Maus|ersatz	Maus|ersatz
Maus|mutanten	Maus|mutanten
Maus|knopf	Maus|knopf
Maus|steuerung	Maus|steuerung
Maus|bewegung	Maus|bewegung
Maus|gene	Maus|gene
Maus|klicks	Maus|klicks
Maus|hirns	Maus|hirns
Maus|zeiger	Maus|zeiger
Maus|hirnen	Maus|hirnen
Maus|bedienung	Maus|bedienung
...	...
(up to 500)	(up to 500)

Extracting prototypes

Maus|kostüm	Maus|kostüm
Maus|zeiger	Maus|zeiger
Maus|stämme	Maus|stämme
Maus|klick	Maus|klick
Maus|hirn	Maus|hirn
Maus|tasten	Maus|tasten
Maus|ersatz	Maus|ersatz
Maus|mutanten	Maus|mutanten
Maus|knopf	Maus|knopf
Maus|steuerung	Maus|steuerung
Maus|bewegung	Maus|bewegung
Maus|gene	Maus|gene
Maus|klicks	Maus|klicks
Maus|hirns	Maus|hirns
Maus|zeiger	Maus|zeiger
Maus|hirnen	Maus|hirnen
Maus|bedienung	Maus|bedienung
...	...
(up to 500)	(up to 500)

Extracting prototypes

Maus|kostüm	Maus|kostüm
Maus|zeiger	Maus|zeiger
Maus|stämme	Maus|stämme
Maus|klick	Maus|klick
Maus|hirn	Maus|hirn
Maus|tasten	Maus|tasten
Maus|ersatz	Maus|ersatz
Maus|mutanten	Maus|mutanten
Maus|knopf	Maus|knopf
Maus|steuerung	Maus|steuerung
Maus|bewegung	Maus|bewegung
Maus|gene	Maus|gene
Maus|klicks	Maus|klicks
Maus|hirns	Maus|hirns
Maus|zeiger	Maus|zeiger
Maus|hirnen	Maus|hirnen
Maus|bedienung	Maus|bedienung
...	...
(up to 500)	(up to 500)

Extracted prototypes for *Maus-*

Prototype	Evidence words
$v_{\text{-Zeiger}}$	-Bewegung -Klicks -Klick -Tasten -Zeiger
$v_{\text{-Stämme}}$	-Mutanten -Gene -Hirnen -Stämme
$v_{\text{-Kostüm}}$	-Knopf -Hirn -Hirns -Kostüm
$v_{\text{-Steuerung}}$	-Ersatz -Bedienung -Steuerung

Compound splitting: *Mausmutation*

Mausmutation

- We start from the left...

Compound splitting: *Mausmutation*

Mausmutation

$\longrightarrow$

- Do I know the modifier *Mau*? No!

Compound splitting: *Mausmutation*

Maus mutation

$\longrightarrow$

- Do I know the modifier *Maus*? Yes!

Compound splitting: *Mausmutation*

Mausmutation

$\longrightarrow$

- Do I know the modifier *Maus*? Yes!

  Prototypes:

  - -Zeiger
  - -Stämme
  - -Kostüm
  - -Steuerung

Compound splitting: *Mausmutation*

Mausmutation

$\longrightarrow$

- Do I know the modifier *Maus*? Yes!

  Prototypes:

  – ~~-Zeiger~~
  – -Stämme ✓ $\rightarrow$ ***Mausmutation*** is to ***Mutation*** what ***Mausstämme*** is to ***Stämme***.
  – ~~-Kostüm~~
  – ~~-Steuerung~~

Compound splitting: *Mausmutation*

Mausmutation

$\longrightarrow$

- Do I know the modifier *Mausm*? No!

Compound splitting: *Mausmutation*

Mausmutation

- And so on...

Compound splitting: *Mausmutation*

Maus|mutation

- The prototype with the highest score will be our split!
- Recurse...

Compound splitting: *Plantage*

Plantage

- Let's try another example...

Compound splitting: *Plantage*

$$\underline{\text{Plan}}\text{tage}$$
$$\longrightarrow$$

- Do I know the modifier *Plan*? Yes!

Compound splitting: *Plantage*

Plantage

$\longrightarrow$

- Do I know the modifier *Plan*? Yes! Prototypes:
  - -Feststellung
  - -Wert
  - -Fertiger
  - ...

Compound splitting: *Plantage*

Plantage

$\longrightarrow$

- Do I know the modifier *Plan*? Yes! Prototypes:
  - ~~-Feststellung~~
  - ~~-Wert~~
  - ~~-Fertiger~~
  - ...

Compound splitting: *Plantage*

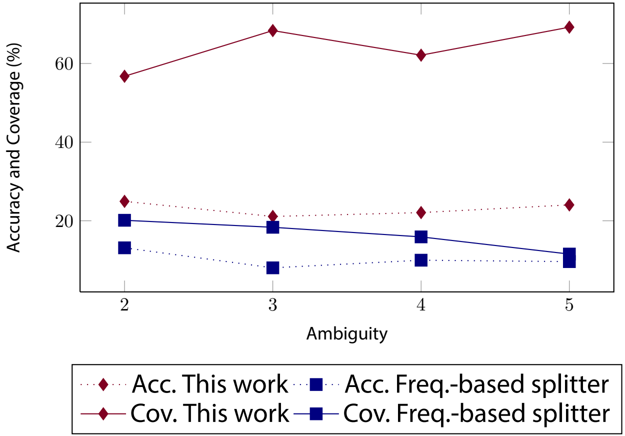
Plantage

- No compound split!

Intrinsic evaluation

- Evaluation on human-annotated dataset (Henrich and Hinrichs, 2011)
  - ~50k compounds
  - only binary splits
- Baseline: Frequency-based Moses compound splitter (Koehn and Knight, 2003)
- We evaluate:
  - Accuracy: $\frac{|\text{correct splits}|}{|\text{compounds}|}$
  - Coverage: $\frac{|\text{compounds split}|}{|\text{compounds}|}$

Intrinsic evaluation



Machine translation experiments (German to English)

	(a) No comp. splitting			(b) Rare: c(w) < 20			(c) All words		
	Splits	BLEU	MTR	Splits	BLEU	MTR	Splits	BLEU	MTR
Moses splitter	0	17.6	25.5	231	17.6	25.7	244	17.9	25.8^{A}
This work				744	$\mathbf{18.2}^{ABC}$	$\mathbf{26.1}^{ABC}$	1616	17.7	26.3^{A}

^{A} Stat. sign. against (a) at $p < 0.05$ ^{B} Stat. sign. against Moses splitter at same c(w) at $p < 0.05$
^{C} Stat. sign. against best Moses splitter (c) at $p < 0.05$

Conclusion

- Regularities in semantic vector space can be used to model composition of compounds
- We can extract modifiers and prototypes (Soricut and Och, 2015)
- Compound splitting algorithm:
  - Good intrinsic performance on gold standard
  - Improved translation quality (standard PBMT setup)
  - Especially adept at splitting highly ambiguous compounds

Thank You!

Any questions?

References

Henrich, V. and Hinrichs, E. W. (2011). Determining immediate constituents of compounds in GermaNet. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing 2011*.

Koehn, P. and Knight, K. (2003). Empirical methods for compound splitting. In *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics-Volume 1*, pages 187--193. Association for Computational Linguistics.

Mikolov, T., Yih, W.-t., and Zweig, G. (2013). Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746--751, Atlanta, Georgia. Association for Computational Linguistics.

Soricut, R. and Och, F. (2015). Unsupervised morphology induction using word embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1627--1637, Denver, Colorado. Association for Computational Linguistics.