Splitting Compounds By Semantic Analogy

Joachim Daiber, Anton Dvorkovich

Institute for Logic, Language and Computation
University of Amsterdam

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

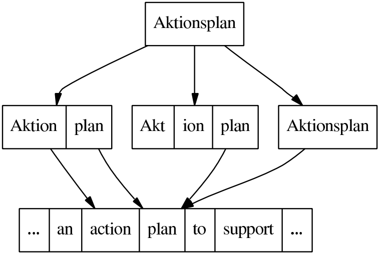

Figure: Compound splitting example from Koehn and Knight (2003).

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)

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:

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

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

- *Maus|zeiger* explains *Maus|klick* **IFF**

$$\textsc{Rank}(v_{\text{cmpd}}, \hat{v}_{\text{cmpd}}) < 100 \quad \textbf{AND} \quad \textsc{Cosine}(v_{\text{cmpd}}, \hat{v}_{\text{cmpd}}) > 0.5$$

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...

© Science Central

Compound splitting: *Mausmutation*

Mausmutation

$\longrightarrow$

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

Compound splitting: *Mausmutation*

$$\underline{\text{Maus}}\text{mutation}$$

$$\longrightarrow$$

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

Compound splitting: *Mausmutation*

Mausmutation

⟶

- 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 ✓ → ***Mausmutation*** is to ***Mutation*** what ***Mausstämme*** is to ***Stämme***.
  - ~~-Kostüm~~
  - ~~-Steuerung~~

Compound splitting: *Mausmutation*

$$\underline{\text{Mausm}}\text{utation}$$

$$\longrightarrow$$

- 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

⟶

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

Compound splitting: *Plantage*

Plantage

- No compound split!

This project

- Better decision: when to split
- Lattice output
- Weights via structured perceptron

Demo

- Demo!

Thank You!

Any questions?

References

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.