



Measuring Semantic Relatedness of German Compounds using GermaNet

Seminar: Lexical-Semantic Processing in NLP

Niels Ott

niels@drni.de

University of Tübingen, Seminar für Sprachwissenschaft



Overview



This presentation will cover the following topics:

- Motivation
- Morphology of compounding
- Consequences for Processing
- Implementation
- Concepts of evaluation
- Open issues



Motivation

- GermaNet contains only a limited number of compounds.
- Compounding is a productive phenomena.
- The number of possible compounds is (virtually) infinite.
- Compounds are widespread.
- We need to handle them somehow!

Design Decision



If a compound is not present in GermaNet, I try to find its parts in GermaNet.

Semantic relatedness is then somehow calculated with these parts.



Morphology



Morphology basically consists of three fields:

- Inflection
- Derivation
- Compounding

Morphology



Morphology basically consists of three fields:

- Inflection
- Derivation
- Compounding
- We are interested only in compounding.



Design Decision



I do not care about derivation and inflection.



Morphology

- Compounds consist of several parts with different parts of speech.

Morphology

- Compounds consist of several parts with different parts of speech.
- A general rule for a (two part) compound would be:

$X \rightarrow Z+X$

Morphology

- Compounds consist of several parts with different parts of speech.
- A general rule for a (two part) compound would be:

$$X \rightarrow Z+X$$

- This can be seen as a modifier+head structure.

Morphology

- Compounds consist of several parts with different parts of speech.
- A general rule for a (two part) compound would be:

$$X \rightarrow Z+X$$

- This can be seen as a modifier+head structure.
- A traditional analysis uses binary branching trees.

Morphology



- In German, there are bounding morphemes, or rather bounding suffixes. (Langer, 1998)
- Examples: *Aktion-s-plan*, *Hund-e-hütte*

Morphology

- In German, there are bounding morphemes, or rather bounding suffixes. (Langer, 1998)
- Examples: *Aktion-s-plan*, *Hund-e-hütte*
- This extends our rule for two part compounds to:

$$X \rightarrow Z+B+X$$

- Bounding suffixes are productive but there are no reliable rules for them.

Examples

$N \rightarrow N + N$ *Eisenbahn, Sommerhut*

$N \rightarrow A + N$ *Kleinholz, Großmaul*

$N \rightarrow V + N$ *Gehweg, Startbahn*

$A \rightarrow A + A$ *dunkelblau, hellgelb*

$A \rightarrow V + A$ *fahrbereit, trinkfest*

$V \rightarrow N + V$ *haushalten, radfahren*

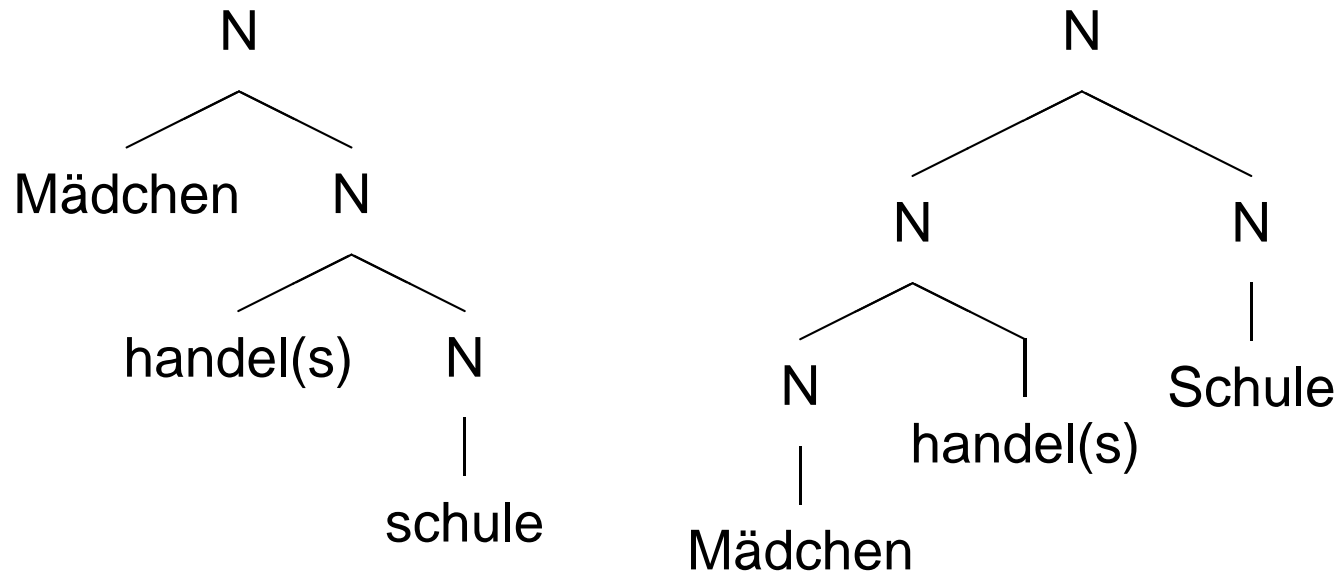
$N \rightarrow P + N$ *Zwischenbericht, Vorschule*

(Bhatt, 1991, p. 40)

Problems



There are bracketing ambiguities. Consider these structures:



How should my computer know the difference?



Problems



The modifier+head assumption often fails from a semantic point of view...

<i>Pferdemarkt</i>	<i>Markt, der Pferde verkauft</i>
<i>Computerproblem</i>	<i>Problem mit einem Computer</i>
<i>Ohrenarzt</i>	<i>Arzt für die Ohren</i>
<i>Profilinguist</i>	<i>Linguist, der Profi ist</i>

The examples above are perfectly logical.



Problems



The modifier+head assumption often fails from a semantic point of view...

<i>Jägerschnitzel</i>	<i>Schnitzel, gemacht aus Jäger?</i>
<i>Froschmann</i>	<i>Männlicher Frosch (Head on the left!) Mann, der ein Frosch ist? (Sense?)</i>
<i>Künstlermarkt</i>	<i>Markt, der Künstler verkauft?</i>
<i>Herzblatt</i>	<i>... ?</i>

These cases are weird!



Problems

- Some compounds work perfectly logically.
- Others require knowledge that is hard to model by software.
- There are linguistic devices that can handle this phenomenon. (Bhatt 1991, p. 44; Langer 1998)
- ... but that's too much for this project.

Design Decision

Compounds consist only of two parts.

(and maybe a bounding suffix)

I call these parts Atoms – they are not divisible.

An Atom may be any entry in GermaNet – even a compound itself.

Weighting Atoms

How important should the parts of a compound be?

Concept 1: The heads are more important than the modifiers.

Simplified for the comparison of a compound W_1 and a normal word W_2 :

$$rel(W_1, W_2) = \frac{1 \cdot rel(A_1, W_2) + \dots + n \cdot rel(A_n, W_2)}{1 + \dots + n}$$

(Building a weighted average.)

Weighting Atoms



There are some problems with this approach:

- From a semantic point of view, the head often is not the major concept!
- *Pferdemarkt* is a kind of *Markt*,
- ...but *Froschmann* is either a male frog or a diver.
- *Herzblatt* is none of its Atoms.



Weighting Atoms



How important should the parts of a compound be?

Concept 2: All Atoms are equally weighted.

Simplified for the comparison of a compound W_1 and a normal word W_2 :

$$rel(W_1, W_2) = \frac{rel(A_1, W_2) + \dots + rel(A_n, W_2)}{count(A)}$$

(Arithmetic mean)



Design Decision



All Atoms are equally weighted.

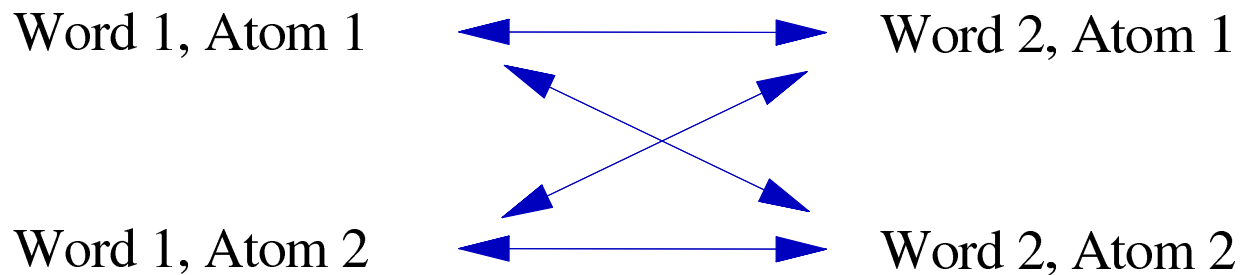
(I hope that this works well enough.)



Crossover Relatedness



This leads us to the combination of 4 distance measurements:

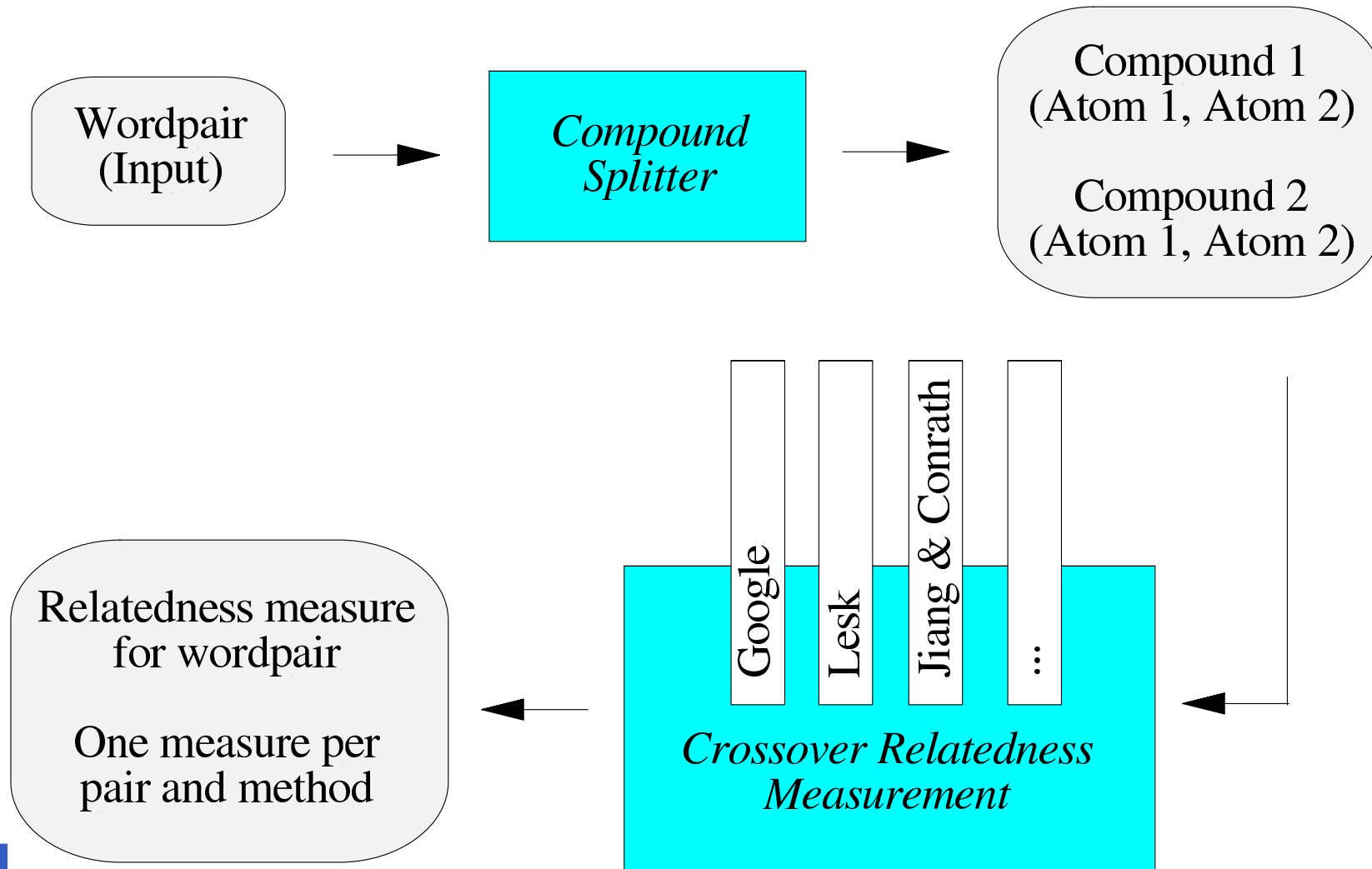


With an arithmetic mean:

$$rel(W_1, W_2) = \frac{1}{4} \cdot (rel(A_{w_1,1}, A_{w_2,1}) + rel(A_{w_1,1}, A_{w_2,2}) + rel(A_{w_1,2}, A_{w_2,1}) + rel(A_{w_1,2}, A_{w_2,2}))$$



Implementation



Compound Splitting

- Most compound splitters work using a dictionary resource.
- They split by all possible full words from the dictionary, plus bounding suffixes:

Aktionsplan

Aktion#s#Plan

Akt#lon#s#Plan

Compound Splitting



- GermaNet can be used as a dictionary resource.



Compound Splitting



- GermaNet can be used as a dictionary resource.
- Simplified sledgehammer approach:

Atom1#BoundingSuffix#Atom2



Compound Splitting

- GermaNet can be used as a dictionary resource.
- Simplified sledgehammer approach:

Atom1#BoundingSuffix#Atom2

- The bounding suffixes are not arbitrary.
(See Langer (1998) for basis of implementation.)

Compound Splitting



- GermaNet can be used as a dictionary resource.
- Simplified sledgehammer approach:

Atom1#BoundingSuffix#Atom2

- The bounding suffixes are not arbitrary.
(See Langer (1998) for basis of implementation.)
- We can not split what is not in GermaNet



Compound Splitting



- GermaNet can be used as a dictionary resource.
- Simplified sledgehammer approach:

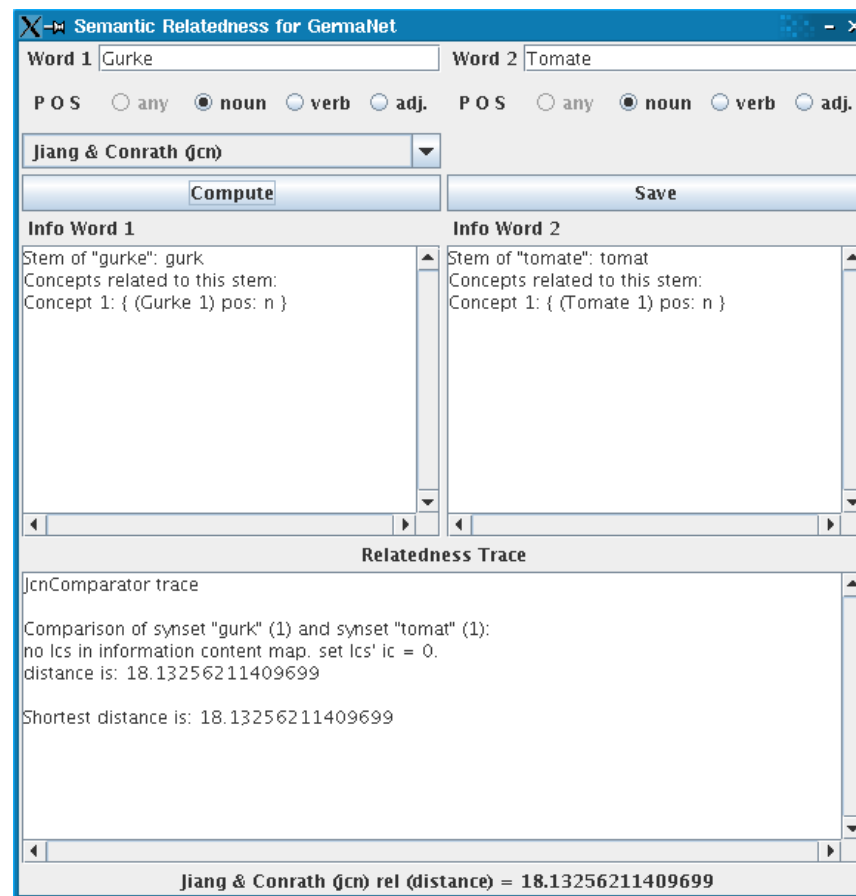
Atom1#BoundingSuffix#Atom2

- The bounding suffixes are not arbitrary.
(See Langer (1998) for basis of implementation.)
- We can not split what is not in GermaNet
- And we can not measure relatedness with what is not in GermaNet, so alright with me!



Relatedness Measurement

- The backends for relatedness measurements are exactly the same as in the SIR_author tool.



Relatedness Measurement



- The backends for relatedness measurements are exactly the same as in the SIR_author tool.
- Difference: Relatedness is measured using an intermediate level that applies the crossover relatedness of Atoms.



Evaluation



There are no results from evaluation yet. Some concepts and thoughts:



Evaluation



There are no results from evaluation yet. Some concepts and thoughts:

- Take Google measures as the baseline.
- Make measures with the program.
- Rate the measures as correlation to the gold standard.



Evaluation



There are no results from evaluation yet. Some concepts and thoughts:

- Open issue: Does the program maybe perform worse on non-compounds than existing tools do?
- This should possibly be taken into account during evaluation.



Open Issues



- GermaNet coverage: Many Atoms are not present, the compound splitter fails.
- Crossover relatedness:
Some measurement methods work only if the compared Atoms have the same POS.
- Weighting of crossover relatedness:
Is the arithmetic mean good enough?
- Compound Splitting: Several issues...
- ... and others to be found by evaluation.



Thank You!







This is the end. Questions?



Acknowledgments

My project makes use of several resources:

-  Java implementation of the named relatedness measures for GermaNet provided by European Media Lab, Heidelberg. <http://www.eml-research.de>
-  GermaNet developed at and provided by the Seminar für Sprachwissenschaft, Tübingen University. <http://www.sfs.nphil.uni-tuebingen.de/lsd/>
-  The Google API for accessing Google. <http://www.google.com/apis/>
-  Some useful free Java classes from Wutka Consulting Inc. <http://www.wutka.com/>

References

- Baroni, Marco, Matiassek, Johannes, and Trost, Harald (2002). *Predicting the components of German nominal compounds*. In van Harmelen, Frank, ed., *Proceedings of the 15th European Conference on Artificial Intelligence (ECAI)*, pp. 470–474. Amsterdam: IOS Press.
URL <http://sslmit.unibo.it/~baroni/publications/E0411.pdf>
- Bhatt, Christa (1991). *Einführung in die Morphologie*. Hürth-Efferen: Gabel, 2nd edn.
- Gurevych, Iryna (2005). *Lexical Semantic Processing in NLP*. Lecture Slides.
URL <http://www.eml-research.de/english/homes/gurevych/teaching/WS2005Plan.php>
- Gurevych, Iryna and Niederlich, Hendrik (2005). *Computing Semantic Relatedness in German with Revised Information Content Metrics*. In *Proceedings of "OntoLex 2005 - Ontologies and Lexical Resources" IJCNLP'05 Workshop*. Jeju Island, Republic of Korea.
URL <http://www.eml-research.de/english/homes/gurevych/downloads/ijcnlp-05.pdf>
- Kunze, Claudia and Lemnitzer, Lothar (2002). *GermaNet - representation, visualization, application*. In *Proceedings: Third International Conference on Language Resources and Evaluation, LREC 2002*, pp. 1485–1491.
URL http://www.sfs.uni-tuebingen.de/~lothar/publ/LREC_main_paper.ps
- Köhn, Phillipp and Knight, Kevin (2003). *Empirical Methods for Compound Splitting*. In *EACL 2003*.
URL <http://people.csail.mit.edu/people/koehn/publications/compound2003.pdf>
- Langer, Stefan (1998). *Zur Morphologie und Semantik von Nominalkomposita*. In *Tagungsband der 4. Konferenz zur Verarbeitung natürlicher Sprache, KONVENS*, pp. 83–97.
URL <http://citeseer.ist.psu.edu/496465.html>