Towards Unsupervised and Language-independent Compound Splitting using Inflectional Morphological Transformations

◊Patrick Ziering ♣Lonneke van der Plas
◊Institute for Natural Language Processing
University of Stuttgart, Germany
♣Institute of Linguistics
University of Malta, Malta

NAACL-HLT 2016
14th June 2016
Kraftfahrzeughafpflichtversicherung
Kraftfahrzeug|haftpflichtversicherung
‘motor car liability insurance’

Kraft|fahrzeug
‘force’

Fahr|zeug
‘vehicle’

fahren
‘to drive’

Zeug
‘stuff’

Haftpflicht|versicherung
‘liability insurance’

Haft|pflicht
‘liability’

Pflicht
‘obligation’

Versicherung
‘insurance’
Compounds

- A **compound** is a new lexeme formed by adjoining two or more lexemes [Bauer, 2003]
  - dog food
  - computer science

- Many languages conflate lexemes into a **closed compound**
  - Weihnachtsbaum: ‘Christmas tree’
  - appelsap: ‘apple juice’
  - fönsterputs: ‘window cleaner’
  - korthus: ‘house of cards’

- database, network
Many applications benefit from compound splitting and normalization (e.g., SMT)

\[
\text{Deutsch} \quad \text{Weihnachtsbaum} \quad \rightarrow \quad \text{Weihnachts} \mid \text{baum} \\
\quad \rightarrow \quad \text{Weihnachten} + \text{Baum}
\]

In order to find the normalized constituents, a compound splitter needs knowledge about linking operations

**suffixation**

\[
\text{Nederlands} \quad \text{kinderboek} \\
\rightarrow \quad \text{kind} + \text{boek}
\]

\[
\text{‘children’s book’}
\]

**truncation**

\[
\text{Deutsch} \quad \text{Wohn_haus} \\
\rightarrow \quad \text{wohnen} + \text{Haus}
\]

\[
\text{‘apartment building’}
\]
Compound splitting

- Compound splitters usually include small sets of hand-crafted rules for most linking operations [Koehn and Knight, 2003]
  - language-dependent
  - need support by language experts

- Automatic learning of linking operations from parallel data [Macherey et al., 2011]
  - need parallel corpora including English
    - sparse and often domain-specific (e.g., EuroParl)
  - rely on word-alignment quality
Rationale for automatic constituent normalization

- Linking operations: only minor changes (at most 3 characters)
- Pure string similarity (e.g., edit distance) is misleading
  - Hühner ‘chicken-’ → Hüne ‘giant’ (ED: 2) ❌
  - → Huhn ‘chicken’ (ED: 3) ✓
- Linguistic restrictor: Word inflection
  → linking operations strongly conform with word inflection
  → verbal conjugation, pluralization, case marker, ...

Our compound splitting approach

- learns linking operations from word inflection
  ⇒ no need for hand-crafted rules
- has no dependence on parallel corpora
We represent string operations as a sequence of replacements

⇒ Morphological Operation Pattern (MOP)

Learning MOPs from word inflection

1 **Input:** Lemmatized tokens (i.e., \(\langle\text{lemma}, \text{word-form}\rangle\) pairs)
2 **Process:** Backtrace of Levenshtein edit distance (ED)
3 **Output:** Grouped ED replacement operations

\[
\text{freq}(\text{MOP}) \equiv \text{freq}(\text{lemmatization})
\]

⇒ inflectional MOPs

| Language | MOP       | Examples                                      |
|----------|-----------|-----------------------------------------------|
| German   | u/ü:$$/er$ | \(\langle\text{Huhn, Hühner}\rangle\) ‘chicken’, \(\langle\text{Buch, Bücher}\rangle\) ‘book’ |
|          | um$/en$   | \(\langle\text{Studium, Studien}\rangle\) ‘study’, \(\langle\text{Medium, Medien}\rangle\) ‘medium’ |
Morphological Operation Pattern

- We represent string operations as a sequence of replacements

  ⇒ Morphological Operation Pattern (MOP)

Learning MOPs from word inflection

1. **Input:** Lemmatized tokens (i.e., <lemma, word-form> pairs)
2. **Process:** Backtrace of Levenshtein edit distance (ED)
3. **Output:** Grouped ED replacement operations

  → \( \text{freq}(\text{MOP}) \equiv \text{freq}(\text{lemmatization}) \)

  ⇒ inflectional MOPs

| Language | MOP | Examples |
|----------|-----|----------|
| German   | u/ü:$/er$ | <Huhn, Hühner> ‘chicken’, <Buch, Bücher> ‘book’ |
|          | um$/en$   | <Studium, Studien> ‘study’, <Medium, Medien> ‘medium’ |
Morphological Operation Pattern

- We represent string operations as a sequence of replacements

⇒ Morphological Operation Pattern (MOP)

**Learning MOPs from word inflection**

1. **Input:** Lemmatized tokens (i.e., \(<\text{lemma}, \text{word-form}>\) pairs)
2. **Process:** Backtrace of Levenshtein edit distance (ED)
3. **Output:** Grouped ED replacement operations

⇒ freq(MOP) ≡ freq(lemmatization)

⇒ *inflectional MOPs*

| Language | MOP   | Examples                                                                 |
|----------|-------|---------------------------------------------------------------------------|
| German   | u/ü:$/er$ | <Huhn, Hühner> ‘chicken’, <Buch, Bücher> ‘book’                           |
|          | um$/en$  | <Studium, Studien> ‘study’, <Medium, Medien> ‘medium’                     |
**Goal:** Allow for all possible linking operations when looking up the composed lemmas during splitting

**Complexity:** We cannot inspect all corpus lemmas for normalization

**Solution:** Ngram index

| Ngram   | Lemma length | Lemmas                      |
|---------|--------------|-----------------------------|
| ^hund   | 4            | hund#13162                  |
|         |              | ‘dog’                       |
|         |              | ‘dog_handler’,              |
|         |              | ‘dog_owner’,                |
|         |              | ‘dog_license’               |
| ^hund   | 11           | hundeführer#251,            |
|         |              | hundehalter#81,            |
|         |              | hundesteuer#64             |
| ^h*hn   | 4            | hahn#2078,                 |
|         |              | huhn#1839,                 |
|         |              | hohn#506                   |
|         |              | ‘rooster’,                  |
|         |              | ‘chicken’,                  |
|         |              | ‘scorn’                     |
**Goal:** Allow for all possible linking operations when looking up the composed lemmas during splitting

**Complexity:** We cannot inspect all corpus lemmas for normalization

**Solution:** Ngram index

| Ngram | Lemma length | Lemmas                                      |
|-------|--------------|---------------------------------------------|
| ^hund | 4            | hund#13162, ‘dog’                           |
|       |              | hundeführer#251, ‘dog_handler’              |
|       |              | hundehalter#81, ‘dog_owner’                 |
|       |              | hundesteuer#64, ‘dog_license’               |
| ^h*hn | 4            | hahn#2078, ‘rooster’                         |
|       |              | huhn#1839, ‘chicken’                         |
|       |              | hohn#506, ‘scorn’                            |
**Goal:** Allow for all possible linking operations when looking up the composed lemmas during splitting

**Complexity:** We cannot inspect all corpus lemmas for normalization

**Solution:** \( N \)gram index

| \( N \)gram | Lemma length | Lemmas | ‘dog’          |
|-------------|--------------|--------|----------------|
| \( ^{\text{hund}} \) | 4            | hund\#13162  | ‘dog’          |
| \( ^{\text{hund}} \) | 11           | hundeführer\#251, hundehalter\#81, hundesteuer\#64 | ‘dog’          |
| \( ^{\text{h*hn}} \) | 4            | hahn\#2078, huhn\#1839, hohn\#506 | ‘rooster’      |
|             |              |        | ‘chicken’      |
|             |              |        | ‘scorn’        |
**Goal:** Allow for all possible linking operations when looking up the composed lemmas during splitting

**Complexity:** We cannot inspect all corpus lemmas for normalization

**Solution:** Ngram index

| Ngram | Lemma length | Lemmas                                      |  |
|-------|--------------|---------------------------------------------|---|
| ^hund | 4            | hund#13162, 'dog'                           |   |
| ^hund | 11           | hundeführer#251, hundehalter#81, hundesteuer#64, 'dog_handler', 'dog_owner', 'dog_license' |   |
| ^h*hn | 4            | hahn#2078, huhn#1839, hohn#506, ‘rooster’, ‘chicken’, ‘scorn’ |   |
**Goal:** Allow for all possible linking operations when looking up the composed lemmas during splitting

**Complexity:** We cannot inspect all corpus lemmas for normalization

**Solution:** Ngram index

| Ngram | Lemma length | Lemmas                                                                 |
|-------|--------------|----------------------------------------------------------------------|
| ^hund | 4            | hund#13162, ‘dog’                                                    |
| ^hund | 11           | hundeführer#251, hundehalter#81, hundesteuer#64, ‘dog_handler’, ‘dog_owner’, ‘dog_license’ |
| ^h*hn | 4            | hahn#2078, huhn#1839, hohn#506, ‘rooster’, ‘chicken’, ‘scorn’         |
Our splitting algorithm

Recursive splitting algorithm

- Binary splitter
  - generates all possible splits
  - Lemma lookup
    - Lemma score
  - Constituent normalization
  - Best split determination
    - Geometric mean * length
    - Head PoS equality

modifier + head

Recursive process

Closed compound

Split tree structure

PoS-tagged

lemmatized

monolingual

corpus

Ngram
index

MOP
frequency

table

PoS
model
Our splitting algorithm

PoS-tagged and lemmatized monolingual corpus (e.g., Wikipedia)
Our splitting algorithm

- Ngram index
- Inflectional MOP corpus frequencies
- PoS probability model ($P(\text{PoS}|\text{word})$)
**Our splitting algorithm**

- **Input:** closed compound (e.g., *Hühnersuppe* ‘chicken soup’)
- Recursive process of a binary splitter
- **Output:** Split tree structure
Our splitting algorithm

- Generates all possible splits (incl. non-split option)

Hühnersuppe, ..., Hühnersuppe, Hühnersuppe
Our splitting algorithm

- Hühner ⇒ ^h*h⋅h ⇒ Hahn, Huhn, Hohn
- score(lemma) = \( \text{freq(lemma)} \cdot \text{freq( MOP\_lemma \rightarrow constituent )} \)
- score(Huhn) = \( \text{freq(Huhn)} \cdot \text{freq( u/ü:$$/er$ )} \)
Our splitting algorithm

- geometric-mean( score(lemmas) · length(constituents) )

\[
geoLen(Huhn,Suppe) = \sqrt{score(Huhn) \cdot 6 \cdot score(Suppe) \cdot 5}
\]
Our splitting algorithm

- Output: best split (lemma pair and split point)

\[ \text{Hühnersuppe} \rightarrow \text{Hühner|suppe} \quad (\text{Huhn, Suppe}) \]

- If binary split: recursive run on modifier and head
Example tree output with inflectional MOPs

Studienbescheinigungssablaufdatum
‘enrollment certification expiration date’

Studienbescheinigung
‘enrollment certification’

Studium
‘study’

Bescheinigung
‘certification’

Ablauf datum
‘expiration date’

ablaufen
‘to expire’

Datum
‘date’

en$/$/

$/$en$
**Experimental setup**

### Monolingual corpora

| Language | Corpus                      | Tokens   |
|----------|-----------------------------|----------|
| 🇩🇪 / 🇳🇱 | Wikipedia                   | 665M / 114M tokens |
| 🇿🇦      | Taalkommissie corpus        | 57M      |

### Compound gold standard

| Language  | Source                                  | Size  |
|-----------|-----------------------------------------|-------|
| 🇩🇪 German | [Henrich and Hinrichs, 2011]            | 51K   |
| 🇳🇱 Dutch  | [Verhoeven et al., 2014]                | 22K   |
| 🇿🇦 Afrikaans | [Verhoeven et al., 2014]            | 17K   |

### Evaluation measures

- Split point accuracy (**SPAcc**): split points
- Normalization accuracy (**NormAcc**): split points and lemmas
Experimental setup

Systems in comparison

Previous work:
- [Fritzinger and Fraser, 2010] (lexicon-based)
- [Weller and Heid, 2012] (hand-crafted rules)
- [Verhoeven et al., 2014] (supervised learner)

Variants of our approach:
- **LP.MS\(_{infl}\)**: Our main approach: inflectional MOPs
- **LP.MS\(_{Langer}\)**: Linking MOPs derived from [Langer, 1998]
- **LP.MS\(_{GS}\)**: Linking MOPs derived from gold standard
- **LP.\(\emptyset\)**: Lemma frequency baseline (no MOP information)
German results for binary compound splitting

| System               | SPAcc    | NormAcc |
|----------------------|----------|---------|
| (A) LP.MS_{infl}     | 95.2%$^{B,C}$ | 86.6%   |
| (B) WH2012           | 94.3%    | 86.8%   |
| (C) FF2010           | 91.4%    | 88.4%$^{A,B}$ |
| (D) LP.∅             | 54.1%    | 28.4%   |
| (E) LP.MS_{Langer}   | 94.5%    | 87.1%   |
| (F) LP.MS_{GS}       | 95.4%    | 87.8%   |

Scores $\delta^{\Phi}$ outperform the system $\Phi$ significantly.
### German results for binary compound splitting

| System                  | SPAcc  | NormAcc |
|-------------------------|--------|---------|
| (A) LP.MS\textsubscript{infl} | 95.2%\textsuperscript{B,C} | 86.6%   |
| (B) WH2012              | 94.3%  | 86.8%   |
| (C) FF2010              | 91.4%  | 88.4%\textsuperscript{A,B} |
| (D) LP.∅                | 54.1%  | 28.4%   |
| (E) LP.MS\textsubscript{Langer} | 94.5%  | 87.1%   |
| (F) LP.MS\textsubscript{GS}   | 95.4%  | 87.8%   |

- scores $\delta^\Phi$ outperform the system $\Phi$ significantly

- LP.MS\textsubscript{infl} outperforms previous work significantly in SPAcc
German results for binary compound splitting

| System             | SPAcc   | NormAcc |
|--------------------|---------|---------|
| (A) LP.MS\textsubscript{infl} | 95.2\%\textsuperscript{B,C} | 86.6\%  |
| (B) WH2012         | 94.3\%  | 86.8\%  |
| (C) FF2010         | 91.4\%  | 88.4\%\textsuperscript{A,B} |
| (D) LP.∅           | 54.1\%  | 28.4\%  |
| (E) LP.MS\textsubscript{Langer} | 94.5\%  | 87.1\%  |
| (F) LP.MS\textsubscript{GS}    | 95.4\%  | 87.8\%  |

scores $\delta^\Phi$ outperform the system $\Phi$ significantly

- LP.MS\textsubscript{infl} outperforms previous work significantly in SPAcc
- LP.MS\textsubscript{infl} is $< 2\%$ below FF2010 (lexicon-based) for NormAcc
German results for binary compound splitting

| System                  | SPAcc | NormAcc |
|-------------------------|-------|---------|
| (A) LP.MS$_{\text{infl}}$ | 95.2%$^{B,C}$ | 86.6%   |
| (B) WH2012              | 94.3% | 86.8%   |
| (C) FF2010              | 91.4% | 88.4%$^{A,B}$ |
| (D) LP.∅                | 54.1% | 28.4%   |
| (E) LP.MS$_{\text{Langer}}$ | 94.5% | 87.1%   |
| (F) LP.MS$_{\text{GS}}$ | 95.4% | 87.8%   |

- LP.MS$_{\text{infl}}$ outperforms previous work significantly in SPAcc
- LP.MS$_{\text{infl}}$ is < 2% below FF2010 (lexicon-based) for NormAcc
- Inflectional MOPs are competitive to linking MOPs from
  - [Langer, 1998]
  - Gold standard of [Henrich and Hinrichs, 2011] (upper bound)
German results for binary compound splitting

| System          | SPAcc | NormAcc |
|-----------------|-------|---------|
| (A) LP.MS\textsubscript{infl} | 95.2\%\textsubscript{B,C} | 86.6\% |
| (B) WH2012      | 94.3\% | 86.8\% |
| (C) FF2010      | 91.4\% | 88.4\%\textsuperscript{A,B} |
| (D) LP.∅        | 54.1\% | 28.4\% |
| (E) LP.MS\textsubscript{Langer} | 94.5\% | 87.1\% |
| (F) LP.MS\textsubscript{GS} | 95.4\% | 87.8\% |

scores $\delta^\Phi$ outperform the system $\Phi$ significantly

- LP.MS\textsubscript{infl} outperforms previous work significantly in SPAcc
- LP.MS\textsubscript{infl} is < 2\% below FF2010 (lexicon-based) for NormAcc
- Inflectional MOPs are competitive to linking MOPs from
  - [Langer, 1998]
  - Gold standard of [Henrich and Hinrichs, 2011] (upper bound)
- LP.∅ baseline underperforms heavily in all cases
# Dutch and Afrikaans results on $N$-ary compound splitting

| System                                      | SPAcc  |
|---------------------------------------------|--------|
| (A) LP.MS$_{infl}$                         | 93.4%$^G$ | 84.7% |
| (G) [Verh.et.al.2014]                      | 91.5%  | 88.3%$^A$ |

Scores $\delta^\Phi$ outperform the system $\Phi$ significantly.

- Only SPAcc because gold standard is not consistently normalized.
Dutch and Afrikaans results on $N$-ary compound splitting

| System                | SPAcc  |
|-----------------------|--------|
| (A) LP.MS$_{infl}$    | 93.4%$^G$ | 84.7% |
| (G) [Verh.et.al.2014] | 91.5%  | 88.3%$^A$ |

Scores $\delta^\Phi$ outperform the system $\Phi$ significantly.

- Only SPAcc because gold standard is not consistently normalized.
- LP.MS$_{infl}$ outperforms [Verh.et.al.2014] significantly for Dutch.
Dutch and Afrikaans results on $N$-ary compound splitting

| System                   | SPAcc |
|--------------------------|-------|
| (A) LP.MS$_{infl}$       | 93.4% $^G$ | 84.7% |
| (G) [Verh.et.al.2014]    | 91.5% | 88.3% $^A$ |

Scores $\delta^\Phi$ outperform the system $\Phi$ significantly.

- Only SPAcc because gold standard is not consistently normalized.
- LP.MS$_{infl}$ outperforms [Verh.et.al.2014] significantly for Dutch.
- LP.MS$_{infl}$ performs worse than [Verh.et.al.2014] for Afrikaans.
  - Smallest corpus (57M vs. 665M / 114M)
Conclusion

- Language-independent unsupervised compound splitter
  → Monolingual PoS-tagged and lemmatized corpora

- Experiments on three Germanic languages
  ⇒ Competitive performance to monolingual splitting methods

- MOPs are suitable for modelling linking operations

- Inflectional morphology: practical approximation for compound splitting in Germanic languages
Thank you for your attention!

Any questions?

- Patrick.Ziering@ims.uni-stuttgart.de
- www.ziering.de
- lonneke.vanderplas@um.edu.mt