Open-Domain Word-Level Interpretation of Norwegian
Towards a General Encyclopedic Question-Answering System for Norwegian

Martin Thorsen Ranang
Department of Computer and Information Science (IDI)

PhD Thesis Presentation, February 5th 2010
Outline

Introduction
  Background and Motivation
  A Central Missing Piece

Mapping Norwegian Words and Phrases to WordNet
  My Approach
  Compound Words
  Algorithms
  Some of the Results

TUClopedemia—Proof of Concept
  System Overview
  Examples
Outline

Introduction
  Background and Motivation
  A Central Missing Piece

Mapping Norwegian Words and Phrases to WordNet
  My Approach
  Compound Words
  Algorithms
  Some of the Results

TUClopedia—Proof of Concept
  System Overview
  Examples
Lack of Advanced Norwegian Language Resources

— Languages shared and spoken by a nation’s citizens are woven into their culture, psychology, education and politics (Lambert, 1972; Joseph, 2004).
— Stimulating language usage in the age of globalization.
— Today’s most advanced language technology:
  • Mostly only available for the English language.
  • Example: the question-answering (QA) system *Powerset* (Converse et al., 2008).\(^1\)
— Languages with many speakers, or of great military and/or economical significance are prioritized.

Lack of Advanced Norwegian Language Resources

  - important to establish a Norwegian Human Language Technology (HLT) resource collection.\(^2\)
  - availability of advanced Norwegian language-technological solutions will strengthen both language and culture.

— Major consequences within several areas, including: research, education, business, and culture.

Goal of the TUClopedia Project

— To create an open-domain question answering (open-domain QA) system that automatically acquires knowledge from, and lets users ask questions about, Norwegian encyclopedia articles.

- *Store norske leksikon* (Henriksen, 2003)

— The response from a QA system should ideally be a *concise*, *relevant*, and *correct* answer.

— Not the same as Information Retrieval (IR) systems (like Google Web Search) that “only”

- perform searches in highly indexed material;
- return lists of relevant documents; and
- highlight relevant passages within text snippets.
The Understanding Computer

— TUC is a Natural Language Understanding (NLU) system developed by Tore Amble.
— Earlier systems based on TUC have been applied to several restricted domains. These include
  • BusTUC, an expert system route adviser for the public bus transport in Trondheim (Amble, 2000);
  • GeneTUC, understood and extracted information about gene and protein interactions from medical articles (Sætre, 2006); and
  • ExamTUC, could extract information from an article about the kings of Norway and automatically grade written answers to examination questions based on the same article (Bruland, 2002).
Outline

Introduction
  Background and Motivation
  A Central Missing Piece

Mapping Norwegian Words and Phrases to WordNet
  My Approach
  Compound Words
  Algorithms
  Some of the Results

TUClopedia—Proof of Concept
  System Overview
  Examples
Lexical-Semantic Network

— Text Interpretation, or NLU, systems depend on the availability of several resources:
  • morphological and lexical knowledge (lexicons);
  • syntax knowledge (grammars); and
  • semantic knowledge (ontologies, or semantic networks).

— TUC uses lexical semantic knowledge during parsing to verify that candidate interpretations of the text make sense.

— The earlier TUC-based systems that were applied to narrow, and/or, restricted domains had used manually crafted highly specialized semantic networks.

— To apply TUC to open-domain QA, a much larger lexical-semantic resource was needed . . . for Norwegian.
Norwegian Lexical-Semantic Resources

— When I began my research, no large-scale lexical-semantic resources existed for Norwegian.

— Two other approaches should be mentioned:
  • “Frå ordbok til ordnett” by Nygaard (2006)
    — Does not distinguish between different senses of words.
    — Provides few semantic relations.
  • *Semantic mirrors* by Dyvik (2004).
    — Can distinguish between word senses.
    — Provides a few more semantic relations than Nygaard’s approach.
Both of Nygaard’s and Dyvik’s approaches generate isolated resources, with no connection to other Natural Language Processing (NLP) resources, something that makes them difficult to integrate into other projects.
Outline

Introduction
  Background and Motivation
  A Central Missing Piece

Mapping Norwegian Words and Phrases to WordNet
  My Approach
  Compound Words
  Algorithms
  Some of the Results

TUClopedia—Proof of Concept
  System Overview
  Examples
My Approach

— I have developed and implemented a method for automatically mapping Norwegian simplex words and compounds to semantic concepts in the Princeton WordNet (Miller, 1995; Fellbaum, 1998) that are warranted by the semantics of the Norwegian source words.

— The implementation of the method produces a Norwegian lexical semantic resource, named *Ordnett*. 
Concepts in *WordNet* are defined by synonym sets (synsets).
Resources

1. A (machine readable) dictionary containing translations from the source language to the target language;
2. A dictionary containing translations from the target language to the source language; and
3. The target ontology of the mapping, a lexical semantic resource for the target language.
4. A lexicon containing morphological information about words in the source language.

(Haslerud and Henriksen, 2003; Fellbaum, 1998; Nordgård, 1998)
Outline

Introduction
  Background and Motivation
  A Central Missing Piece

Mapping Norwegian Words and Phrases to WordNet
  My Approach
  Compound Words
  Algorithms
  Some of the Results

TUClopedia—Proof of Concept
  System Overview
  Examples
Automatic Analysis of Compound Words

— «slottsvinduer» (“castle windows”):
  • «slott» (“castle”)
  • «svin» (“pig”)
  • «vin» (“wine”)
  • «vind» (“wind”)
  • «du» (“you”)
  • «due» (“dow”)
  • «duer» (“dows”)
  • «er» (“is”)

Automatic Analysis of Compound Words

<word>
  |<compound word>
  |<compound noun>
  
  |<left stem>
  |<stem>
  |<non monosyllabic noun stem>
  |"musikk"

<word>
  |<compound word>
  |<compound noun>
  
  |<left part>
  |<right part>
  |<left stem>
  |<noun stem>
  |<non monosyllabic noun stem>
  |"spiller"

<word>
  |<compound word>
  |<compound noun>
  
  |<left part>
  |<right part>
  |<left stem>
  |<epenthetic s>
  |<noun>
  |<non sibilant ending noun stem>
  |"musikk"
  |"s"
  |"piller"
Results

Tested on 5 randomly chosen Norwegian newspaper articles (4951 words, 109 compounds) with a 100 % success rate.
Outline

Introduction
  Background and Motivation
  A Central Missing Piece

Mapping Norwegian Words and Phrases to WordNet
  My Approach
  Compound Words

Algorithms

Some of the Results

TUClopedica—Proof of Concept
  System Overview
  Examples
Informal Description of the Main Principle

— Translating the Norwegian word «skatt», we get, e.g., “tax”, “treasure”, and “honey”. Each with multiple senses.
— “Honey” has at least two senses:
   1. “a sweet yellow liquid produced by bees”
   2. “a beloved person”
— Only the second sense above is warranted by the Norwegian word «skatt».
— The main principle is to exploit the semantic information already in WordNet to remove unwarranted mappings.
— For example, by inverse translating the complement synonyms of “honey”, like “loved one” or “beloved”, we can see look for overlaps between the inverse translations and the original word.
Basic Algorithm I

1: procedure MAP-WORD-TO-SENSE(w, C, D_{\ell}, D_{\ell}')
2:   mappings ← ∅
3:   for all t ∈ TRANSLATE(w, C, D_{\ell}') do
4:     T ← SENSES-OF(t, C)
5:     if |T| = 1 then
6:       mappings ← mappings ∪ {(w, T[0])}
7:       continue
8:   for all t_i ∈ T do
9:     mappings ← mappings ∪ MIRROR(w, t_i, C, D_{\ell}')
10:  return mappings

TRANSLATE("brystkasse", n, NorEng) =
{“chest”, “rib cage”, “thorax”}

SENSES-OF("chest", n) =
⟨chest_n_1, chest_n_2, chest_n_3⟩
**Basic Algorithm II**

1: procedure $\text{MIRROR}(w, t_i, C, D_{\ell'})$
2: \hspace{1em} mappings $\leftarrow \emptyset$
3: \hspace{1em} $S \leftarrow \text{GET-SYNSET}(t_i)$
4: \hspace{1em} $\overline{S} \leftarrow S - \{t_i\}$
5: \hspace{1em} for all $s \in \overline{S}$ do
6: \hspace{2em} $w' \leftarrow \text{WORD-FROM-SENSE}(s)$
7: \hspace{2em} for all $w \in \text{TRANSLATE}(w', C, D_{\ell'})$ do
8: \hspace{3em} if $w = w$ then
9: \hspace{4em} mappings $\leftarrow$ mappings $\cup \{(w, t_i)\}$
10: \hspace{1em} return mappings

GET-SYNSET(chest_n_1) = 
{thorax_n_2, chest_n_1, pectus_n_1}

$\overline{S} \leftarrow \{\text{thorax}_n_2, \text{pectus}_n_1\}$

TRANSLATE("thorax", n, EngNor) = 
{«bryst», «brystkasse»,
«toraks», «kropp», «forkropp»}. 

www.ntnu.no Martin Thorsen Ranang, Open-Domain Word-Level Interpretation of Norwegian
Extended Algorithm

1: procedure MIRROR\( (w, t_i, C, D_{\ell'}) \)
2:   mappings \leftarrow \emptyset
3:   S \leftarrow \text{GET-SYNSET}(t_i)
4:   for all \text{STRATEGY} \in \langle \text{SYNONYM}, \text{HYPERNYM} \rangle \text{ do}
5:     \overline{S} \leftarrow \text{STRATEGY}(S, t_i) \quad \triangleright \text{The “complement” synset.}
6:     for all \( s \in \overline{S} \text{ do} \quad \triangleright \text{Complement senses.}
7:       w' \leftarrow \text{WORD-FROM-SENSE}(s)
8:       for all \( w \in \text{TRANSLATE}(w', C, D_{\ell'}) \text{ do}
9:         if \( w = w' \text{ then}
10:           \text{mappings} \leftarrow \text{mappings} \cup \{ (w, t_i) \}
11:       \text{if mappings} \neq \emptyset \text{ then}
12:         \text{break}
13:   \text{return mappings}
Typical Size of a Smaller Case
Outline

Introduction
  Background and Motivation
  A Central Missing Piece

Mapping Norwegian Words and Phrases to WordNet
  My Approach
  Compound Words
  Algorithms
  Some of the Results

TUClopedi—Proof of Concept
  System Overview
  Examples
## Ordnett—Coverage

**Table**: Coverage results of running Verto with the extended dictionaries and the SYNONYM and all of the search strategies.

<table>
<thead>
<tr>
<th>Lexical category</th>
<th>Keywords</th>
<th>Mapped</th>
<th>No inverse</th>
<th>Mappings</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$N$</td>
<td>(%)$^a$</td>
<td>$N$</td>
</tr>
<tr>
<td>Noun</td>
<td>45,993</td>
<td>25,879</td>
<td>56.3</td>
<td>15,783</td>
</tr>
<tr>
<td>Adjective</td>
<td>9,526</td>
<td>5,357</td>
<td>56.2</td>
<td>3,167</td>
</tr>
<tr>
<td>Verb</td>
<td>5,435</td>
<td>4,264</td>
<td>78.5</td>
<td>1,010</td>
</tr>
<tr>
<td>Adverb</td>
<td>1,394</td>
<td>492</td>
<td>35.3</td>
<td>809</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>62,348</td>
<td>35,992</td>
<td>57.7</td>
<td>20,769</td>
</tr>
</tbody>
</table>

$^a$ Percentage of keywords mapped to WordNet senses.

$^b$ Percentage of cases where *no* mapping could be found.
Ordnett—Precision, Recall, $F$-measure

Table: Precision, recall, and $F_{0.5}$ results of running Verto with the original dictionaries and the SYNONYM and SIMILAR search strategies.

<table>
<thead>
<tr>
<th>Lexical category</th>
<th>Correct</th>
<th>Incorrect</th>
<th>Precision</th>
<th>Recall</th>
<th>$F_{0.5}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noun</td>
<td>129</td>
<td>7</td>
<td>0.949</td>
<td>0.222</td>
<td>0.573</td>
</tr>
<tr>
<td>Adjective</td>
<td>219</td>
<td>10</td>
<td>0.956</td>
<td>0.336</td>
<td>0.698</td>
</tr>
<tr>
<td>Verb</td>
<td>160</td>
<td>26</td>
<td>0.860</td>
<td>0.161</td>
<td>0.460</td>
</tr>
<tr>
<td>Adverb</td>
<td>110</td>
<td>10</td>
<td>0.917</td>
<td>0.308</td>
<td>0.657</td>
</tr>
<tr>
<td>Total</td>
<td>618</td>
<td>53</td>
<td>0.921</td>
<td>0.239</td>
<td>0.587</td>
</tr>
</tbody>
</table>
Ordnett: Hypernymy
Ordnett: Hypernymy And Meronymy

{conveyance_n_3, transport_n_1} → a kind of

befordringsmiddel, kjøretøy, kjøredoning, transportmiddel

doning

{vehicle_n_1} → a kind of

fremkomstmiddel, transport, overføring

{craft_n_2} → a kind of

fartøy, farkost

fly, rutefly → {airplane_n_1, aeroplane_n_1, plane_n_1}

part of

{accelerator_n_1, acclerellor_pedal_n_1, gas_pedal_n_1, gas_n_5, throttle_n_2, gun_n_6}

{car_n_1, auto_n_1, automobile_n_1, machine_n_4, motorcar_n_1}

part of

bilk, vogn

{grille_n_2, radiator_grille_n_1}

grill, jitter

{motor_vehicle_n_1, automotive_vehicle_n_1} → a kind of

motorvogn

{bicycle_n_1, bike_n_2, wheel_n_7, cycle_n_6}

a kind of

{container_n_1}

beholder, boks, container, form, kar, kasse, tank, vekselflak

sykkel, tohjulssykkel, träsykkel

motorkjøretøy

NTNU
Norwegian University of Science and Technology
Ordnett: Meronymy And Entailment

hår, pels

{hair_n_1}

substance of

{keratin_n_1, ceratin_n_1}

keratin

substance of

{feather_n_1, plume_n_2, plumage_n_1}

fjær, fjærdrakt

sluke, svelge

{swallow_v_1, get_down_v_4}

entails

innta → {eat_v_1}

entails

cchas_v_1, masticate_v_2, manducate_v_1, jaw_v_3}

ete, fortære, nyte, spise

NTNU
Norwegian University of Science and Technology
Outline

Introduction
  Background and Motivation
  A Central Missing Piece

Mapping Norwegian Words and Phrases to WordNet
  My Approach
  Compound Words
  Algorithms
  Some of the Results

TUClopedia—Proof of Concept
  System Overview
  Examples
System Overview

User interface

Translation framework

Lexical analyzer

Named-entity recognizer

Named-entity extractor

Bilingual dictionary

Lexicon (Morphology)

Grammar (Syntax rules)

Semantic network

WordNet

Onomasticon

Encyclopedia

Articles

Named entities

Question

Tagged words

Parse trees

Query expression

Reasoning engine

Acquired knowledge

Knowledge base

NTNU Norwegian University of Science and Technology

www.ntnu.no Martin Thorsen Ranang, Open-Domain Word-Level Interpretation of Norwegian
Outline

Introduction
  Background and Motivation
  A Central Missing Piece

Mapping Norwegian Words and Phrases to WordNet
  My Approach
  Compound Words
  Algorithms
  Some of the Results

TUClopedia—Proof of Concept
  System Overview
  Examples

NTNU
Norwegian University of Science and Technology
A+

programmeringsspråk for datamaskiner, avledet av APL.
A+—Word Graph

txt(w('A+', name('A+', n, programming_language_n_1)), 0, 1).
txt(w('er', verb(be_v_2, pres, fin)), 1, 2).
txt(w('programmeringsspråk',
    noun(programming_language_n_1, plu, u, n)), 2, 3).
txt(w('programmeringsspråk',
    noun(programming_language_n_1, sin, u, n)), 2, 3).
txt(w('for', prep('for')), 3, 4).
txt(w('datamaskiner', noun(computer_n_1, plu, u, n)), 4, 5).
txt(w('avledet', verb(derive_v_3, past, part)), 5, 6).
txt(w('avledet', ['avledet']), 5, 6).
txt(w('av', prep('from')), 6, 7).
txt(w('av', prep('of')), 6, 7).
txt(w('av', prep('off')), 6, 7).
txt(w('APL', name(apl, n, programming_language_n_1)), 7, 8).
txt(w('.', ['.’]), 8, 9).
A+—Acquired Knowledge (TQL) I

% “A+ programmeringsspråk for datamaskiner, avledet av APL.”
% (A+ programming language for computers, derived from APL.)

isa(programming_language_n_1, real, sk(1))
isa(computer_n_1, real, sk(2))

% Programming language, sk(1), is for computer, sk(2).
event(sk(3))
agent(sk(1), sk(3))
present(real, sk(3))
action(be_v_4, sk(3))
mod(for, sk(2), sk(3))

% Somebody (pro) derives programming language, sk(1), from 'APL'
event(sk(4))
present(real, sk(4))
action(derive_v_3, sk(4))
patient(sk(1), sk(4))
agent(pro, sk(4))
mod(from, 'APL', sk(4))

% A+ is programming language, sk(1).
event(sk(5))
agent('A+', sk(5))
present(real, sk(5))
action(be_v_2, sk(5))
patient(sk(1), sk(5))
Drapet på Kennedy

Presidenten ble myrdet av en snikskytter 22. november 1963 under et besøk i Dallas, Texas. Kennedy-mordet ble umiddelbart gjenstand for en rekke ulike teorier og påstander som har fortsatt å oppta allmennheten etterpå. […]  

[The Kennedy Assassination.  
The President was assassinated November 22, 1963 during a visit to Dallas, Texas. The Kennedy assassination immediately became subject to a range of different theories and allegations that subsequently have continued to occupy the general public. […] ]
Acquired Knowledge

\texttt{isa(visit\_n\_5, real, sk(3))}
\texttt{isa(dallas\_n\_1, real, sk(4))}

% There is a visit, sk(3), in Dallas, sk(4).
\texttt{event(sk(5))}
\texttt{agent(sk(3), sk(5))}
\texttt{present(real, sk(5))}
\texttt{action(be\_v\_4, sk(5))}
\texttt{mod(in, sk(4), sk(5))}

\texttt{isa(texas\_n\_1, real, sk(6))}

% The visit, sk(3), is in Texas, sk(6).
\texttt{event(sk(7))}
\texttt{agent(sk(3), sk(7))}
\texttt{present(real, sk(7))}
\texttt{action(be\_v\_4, sk(7))}
\texttt{mod(in, sk(6), sk(7))}

\texttt{isa(president\_n\_3, real, sk(1))}
\texttt{isa(sniper\_n\_1, real, sk(2))}

% The President, sk(1), was murdered by a
% sniper, sk(2), during the visit, sk(3),
% on November 22, 1963.
\texttt{event(sk(8))}
\texttt{past(real, sk(8))}
\texttt{action(murder\_v\_1, sk(8))}
\texttt{patient(sk(1), sk(8))}
\texttt{agent(sk(2), sk(8))}
\texttt{mod(under, sk(3), sk(8))}
\texttt{mod(nil, date(1963, 11, 22), sk(8))}
Question Answering

«Hvem myrdet presidenten?» (“Who murdered the president?”)

```
which(A),
isa(somebody_n_1, real, A),
isa(president_n_3, real, B),
agent(A, C),
action(murder_v_1, C),
past(real, C),
event(C),
patient(B, C)
```

```
which(sk(2))
isa(somebody_n_1, real, sk(2)) <=
  isa(person_n_1, real, sk(2)) <=
    isa(expert_n_1, real, sk(2)) <=
      isa(sniper_n_1, real, sk(2)) % A = sk(2).
isa(president_n_3, real, sk(1)) % B = sk(1).
agent(sk(2), sk(8)) % C = sk(8).
action(murder_v_1, sk(8))
past(real, sk(8))
event(sk(8))
patient(sk(1), sk(8))

=>

sk(2)
```
References I


References III


References IV


References V


Appendix
Other Non-English Languages

— Compounding in, for example, German:
  
  • “usability”:
    
    *der Gebrauch* (usage) + *die Tauglichkeit* (capability)
    
    → *die Gebrauch-s-tauglichkeit*
  
  • “horse paddock”:
    
    *das Pferd* + *die Koppel* → *die Pferd-e-Koppel*
Research Questions and Answers

Q1 Is it possible to develop a method for automatically building a broad-domain, general semantic resource for Norwegian that is compatible with existing freely available, widely used, English semantic resources?

A1 Yes. Verto, the implementation of the novel method and algorithms presented in my thesis, was successfully used to automatically create Ordnett, a broad-domain, general \textit{lexical-semantic} resource that maps Norwegian words and phrases to concepts in the Princeton \textit{WordNet}. 
Research Questions and Answers

Q2 What restrictions apply to the method? What resources are prerequisite?

A2 The method described herein requires (1) a simple bilingual dictionary that contains information about the *lexical category* of entries and maps words in the source language to words in the target language; and (2) a lexical database that contains semantic information about *synonymy*, *hypernymy*, and *similarity* between senses in the target language. To handle compound words, the method also requires the availability of an automatic compound analyzer.
Research Questions and Answers

Q3 For each of the *lexical categories* nouns, verbs, adjectives, and adverbs, how large fraction of the words in each class does the method work for?

A3 With the configuration of the mapping framework that provided the maximum average *precision* value of 0.921 (achieved in Test Run 4), the number of successfully mapped words per lexical category were *noun* 22,888 (50.3 %); *adjective* 4,387 (47.7 %); *verb* 3,495 (65.2 %); and *adverb* 186 (31.9 %). With the configuration that provided the maximum *recall* value of 0.365 (achieved in Test Run 12), the number of successfully mapped words per lexical category were *noun* 25,879 (56.3 %); *adjective* 5,357 (56.2 %); *verb* 4,264 (78.5 %); and *adverb* 492 (35.3 %). The same configuration also provided the maximum $F_{0.5}$-score of 0.680.
Q4 How well does the method avoid mapping words in the source language to senses in the target language that carry meanings that are not covered by the source words?

A4 Comparison of the results produced by the system presented herein with those produced by a human expert showed that with a configuration that maximizes the precision value, the system reaches a precision score of 0.921. This shows that the system is able to rather precisely avoid unwarranted mappings.
Q5 Can the method handle (single word) *compounds* that occur frequently in Norwegian, but more seldom in English?

A5 Yes. The framework that implements the method presented herein handles mapping of compounds by applying an integrated *compound-word analyzer*. 
Research Questions and Answers

Q6  How may the resulting semantic resource be useful for different areas of NLP research and development?
A6  One of the most important benefits of the method is that it produces a semantic resource that is linked to the Princeton WordNet, and thereby can benefit from other semantic resources that extends or complements—but are still compatible with—WordNet.

[...] It should be possible to utilize Ordnett for Norwegian (versions of) applications within most areas of NLP where ontologies are already used. These areas include, but is not necessarily limited to, Word-Sense Disambiguation (WSD), Text Summarization, Document Clustering, Information Extraction (IE), IR, Text Interpretation, and Natural Language Generation (NLG).
Research Questions and Answers

Q7 Can the method be applied to other languages, and if so, to which ones?

A7 I see no reason why the method should not be applicable to at least other *Germanic languages*, like the Scandinavian languages, German, and Dutch. However, if the language makes extensive use of *compounding*, an automatic *compound-word analyzer* may be required.