FDVC
CREATING A CORPUS-DRIVEN FREQUENCY DICTIONARY OF VERB PHRASE CONSTRUCTIONS FOR HUNGARIAN

Bálint Sass and Júlia Pajzs
sass.balint@nytud.hu, pajzs@nytud.hu

Research Institute for Linguistics, Hungarian Academy of Sciences,
Budapest, Hungary

eLexicography in the 21st century Conference
22-24 October 2009, Louvain-la-Neuve
We present a semi-automatic method for creating a special dictionary which . . .

— is a monolingual, corpus-driven [Tognini-Bonelli(2001)], frequency dictionary;
— has *verb phrase constructions* (VPCs) as entries;
— is a *meaningless* dictionary in the sense of [Janssen(2008)];
— is for Hungarian
  but the core methodology is *language independent*;
— is created in a *mostly automatic* way
  with less manual lexicographic work;
— can be created with a low budget;
— can be useful in language teaching and NLP both.
**Framework:**

- lexical acquisition from corpus → raw dictionary
- manual lexicographic work → final dictionary

How far we can get using automatic tools?
How much we can reduce manual lexicographic work?
1. **What does it look like?**

2. **VPC extraction**

3. **Manual lexicographic work**

4. **Example**

5. **Language independency**

6. **Application**
1. **What does it look like?**
2. **VPC extraction**
3. **Manual lexicographic work**
4. **Example**
5. **Language independency**
6. **Application**
We “jettison ruthlessly” [Hanks(2008)] all verbs and constructions which have zero or low frequency in our corpus.

We do not allow the lexicographer to add any “missing” constructions.

We take the most frequent VPCs into account and record and display their frequency in the dictionary.

We focus on frequent patterns and do not “seek to cover all possible meanings and all possible uses” [Hanks(2001)].
**ENTRY: verb phrase constructions** (1/2)

<table>
<thead>
<tr>
<th>sentence</th>
<th>construction</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) “I believe in miracles.”</td>
<td>free <em>in</em>-slot</td>
</tr>
<tr>
<td>(2) “The girl shrugs her shoulder.”</td>
<td>fixed object slot</td>
</tr>
<tr>
<td>(3) “Nobody takes that into consideration.”</td>
<td>free object slot + fixed <em>into</em>-slot</td>
</tr>
</tbody>
</table>

- **Verb phrase construction (VPC):**
  a verb plus some NP/PP dependents/slots

- Free slots correspond to valences.
  Fixed slots plus the verb form a *multiword verb*.

- Some constructions show both properties.
If we take the fixed slots as part of the multiword verb itself, we can treat simple and multiword verbs the same way (both having some valences beside):

**Example**

`to consider ← to take into consideration`

→ e.g. we can directly compare their frequencies, their significance.

Entries in FDVC are VPCs, the microstructure apparently integrates phraseology as the a basic units are phrases.

**Note:** We arrange the VPCs around a verb in a subsequent step to form more traditional dictionary entries.
“Meaningless dictionary” [Janssen(2008)]: no explicit definitions, just VPCs together with corpus frequencies.

Suitable corpus sentence exemplifying the meaning.

This meaning is fairly concrete:

- VPCs being collocations tend to have only one meaning [Yarowsky(1993)].
- Most VPCs are real constructions, “form and meaning pairings” [Goldberg(2006)], as they cannot be broken down into smaller units without loss of meaning.
- Each VPC represent a pattern of use, and can be paired with one sense of its main (simple or multiword) verb.
**How do we proceed?**

- **Corpus**: Hungarian National Corpus [Váradi(2002)]

- **Automatic** steps – using natural language processing tools
  - (morphological analysis, POS tagging)
  - chunking – have verbs and NP/PP dependents
  - a lexical acquisition algorithm to collect frequent VPCs [Sass(2009)]
  - a simple algorithm to collect suitable examples for VPCs

- **Manual** step: lexicographic work
  (error correcting and example selecting)
1. **What does it look like?**

2. **VPC extraction**

3. **Manual lexicographic work**

4. **Example**

5. **Language independency**

6. **Application**
**Motivation**

- Valence-bearing multiword verbs are borderline cases between verb subcategorization frames and multiword expressions.
- Although they are frequent, they usually get out of field of vision of NLP methods.
- Dictionaries should obviously contain them, so we need lexical acquisition methods to handle them also.

We need a lexical acquisition method which can collect *all* kinds of VPCs as we want to have a complete picture of the language in our dictionary.

*Aim:* to develop such a method.
**Corpus representation:**
Clause skeleton (CS) of sentence (3):

```
take SUBJ:nobody OBJ:that into:consideration ← INPUT
```

Its VPC:

```
take SUBJ OBJ into:consideration ← OUTPUT
```

**Main idea:** we store initially all slots and all content words and allow the algorithm to get rid of . . .

1. complete slots, when they are not integral part of a VPC;
2. infrequent content words, where they are just filling in a valence slot.
HOW DOES IT WORK?

1. We take all CSs of the corpus with frequency counts. *Alternating omission:* add some “free” CS variants to the initial list:

<table>
<thead>
<tr>
<th>VPC</th>
<th>length</th>
</tr>
</thead>
<tbody>
<tr>
<td>shrug SUBJ:girl OBJ:shoulder →</td>
<td>4</td>
</tr>
<tr>
<td>shrug SUBJ OBJ:shoulder</td>
<td>3</td>
</tr>
<tr>
<td>shrug SUBJ:girl OBJ</td>
<td>3</td>
</tr>
<tr>
<td>shrug SUBJ OBJ</td>
<td>2</td>
</tr>
</tbody>
</table>

2. We sort the resulting verb frame list according to *length*.

3. Starting with the longest one we discard CSs with frequency less than 5, and add their frequency to a *one-unit-shorter* frame on the list.

4. VPCs are the final remaining verb frames, ranked by cumulative frequency.
**EXPLANATION**

The algorithm is based on cumulative frequency of corpus patterns, it treats fixed and free slots appropriately [Sass(2009)].

“take SUBJ OBJ into:consideration” will be on the resulting list because in the corpus clauses whose main verb is *take*:

- the *into* slot is usually filled by the word *consideration* – so its frequency can cumulate,
- but the OBJ slot is much more variable – so words in this slot are more easily dropped out.
1. What does it look like?

2. VPC extraction

3. Manual lexicographic work

4. Example

5. Language independency

6. Application
**Task of the lexicographer**

Automatic steps supersede a substantial amount of manual lexicographic work.

→ VPCs in XML form – to be edited in an XML editor.

- error correcting
  Does the VPC exist? Is it correct?
  *cause of errors*: imperfect NLP tools

- idiom selecting
  *criterion*: Has the VPC its own separate meaning?

- example selecting, customizing
  "corpus-based" examples
  [Kilgarriff et al.(2008)]’s suggestions: full-sentence examples, or at least clauses with full predicate structure, avoid personal names etc. If none appropriate: use of a special corpus query system [Sass(2008)].

Two passes: first pass + check
Evaluation of the Automatic Part

Question: how many entries are accepted/rejected by the lexicographer?
Lexicographers completed 2712 VPCs (of 1058 verbs) from which they found the following erroneous:
- 136 separate VPCs
- 43 full entries containing 98 VPCs

The accuracy of the automatic steps is:

\[
\frac{2712 - 136 - 98}{2712} = 91.4\%
\]

Note: 250 VPCs are marked as idioms
→ the idiom rate is \( \frac{250}{2478} = 10.0\% \)
INDEXES

Beside the traditional (alphabetically ordered by verb) presentation we plan to have several indexes. All of them can be generated automatically:

- aggregated list of all VPCs sorted by frequency – in fact this is the true FDVC;
- an index by general patterns (i.e. VPCs without the verb);
- an index by number of fixed/free slots;
- a frequency list of verb stems;
- an index by words in fixed slots.
Counts

Using a very high frequency threshold: 250
... we have:

8519 verb phrase constructions (of 2969 verbs)

Type distribution:

<table>
<thead>
<tr>
<th>type</th>
<th>example</th>
<th>count</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 valence</td>
<td>believe in</td>
<td>3181</td>
<td>37%</td>
</tr>
<tr>
<td>2 valences</td>
<td>give OBJ, OBJ$_D$</td>
<td>1563</td>
<td>18%</td>
</tr>
<tr>
<td>bare verb</td>
<td>happen</td>
<td>1469</td>
<td>17%</td>
</tr>
<tr>
<td>1 fixed word</td>
<td>shrug shoulder</td>
<td>1080</td>
<td>13%</td>
</tr>
<tr>
<td>1 fixed word + 1 valence</td>
<td>take OBJ into account</td>
<td>1043</td>
<td>12%</td>
</tr>
<tr>
<td>other</td>
<td></td>
<td>183</td>
<td>2%</td>
</tr>
</tbody>
</table>
The dictionary containing about 3000 verbs and 8500 VPCs altogether has the following costs:

- automatic steps: programming and support costs
  – about 1 man-year
- manual step: lexicographic work
  – about 1 man-year

Checking is important, and relatively time consuming:
inside the lexicographic work . . .

- the first pass is about 6 man-months,
- and the checking is also 6 man-months.
1. **What does it look like?**

2. **VPC extraction**

3. **Manual lexicographic work**

4. **Example**

5. **Language independency**

6. **Application**
EXAMPLE: VPCs of *elver (to beat)* – XML

```xml
<entry>
  <verb lemma="elver" freq="744"/>
  <pattern freq="284">
    <frame><p c="-t"/></frame>
    <type str="1:01" len="1" fixed="0" free="1"/>
    <cits>
      <cit>hogy minap elverteleke azért,</cit>
    </cits>
  </pattern>
  <pattern freq="36">
    <frame><p c="" l="jég"/><p c="-t"/></frame>
    <type str="3:11" len="3" fixed="1" free="1"/>
    <cits>
      <cit type="sentence">Már ahol a jég nem verte el ...</cit>
    </cits>
  </pattern>
</pattern>
...
**Example: VPCs of elver (to beat) – Dict. Entry**

The corresponding dictionary entry showing the most important three verb phrase constructions of this verb is:

elver [744]
elver -t [284] minap elvertelek azért, ...
   elver jég -t [36] Már ahol a jég nem verte el a termést!
elver -n por-t [95] hogy egy pár túlbuzgó helyi tanácselnökön verjék el a port.

English translation of the entry:

beat [744]
beat OBJECT [284] I beat you yesterday, because ...
   beat ice OBJECT [36] Just where the hail did not destroy the crop!
beat ON dust-OBJECT [95] to blame some overzealous local mayors.
<table>
<thead>
<tr>
<th></th>
<th>What does it look like?</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>VPC extraction</td>
</tr>
<tr>
<td>3</td>
<td>Manual lexicographic work</td>
</tr>
<tr>
<td>4</td>
<td>Example</td>
</tr>
<tr>
<td>5</td>
<td>Language independency</td>
</tr>
<tr>
<td>6</td>
<td>Application</td>
</tr>
</tbody>
</table>
Corpus representation, VPC and example collecting algorithms are in essence language independent. In fact, it only relies on the existence of predicate-argument structure.

So the methodology could be applied to other languages as well, if we have a POS tagger and a suitable chunker.
LANGUAGE INDEPENDENCY

Preliminary experiment for Danish.

Danish Dependency Treebank – *small corpus*: multiword verbs does not come out
e.g. få lov til (to be allowed to), have brug for (to need) etc.
Some promising results:

- **se**:
  - se [28]
  - se på [9]

- **komme**:
  - komme [21]
  - komme til [11]
  - komme til at [8]
  - komme i [11]
  - komme på [9]
1. What does it look like?
2. VPC extraction
3. Manual lexicographic work
4. Example
5. Language independency
6. Application
APPLICATION

- for Hungarian grammar-writers, linguists
- useful in Hungarian lexicography
- *language learning*: advanced level
  It allows the students “to write and speak idiomatically” [Hanks(2008)].
  e.g. *requirements* → *meet* is the appropriate verb
- a rich lexical resource
  from which many NLP tasks could benefit
**Application**

- for Hungarian grammar-writers, linguists
- useful in Hungarian lexicography
- *language learning*: advanced level
  It allows the students “to write and speak idiomatically” [Hanks(2008)].
  e.g. *requirements* → *meet* is the appropriate verb
- a rich lexical resource
  from which many NLP tasks could benefit

Thank you for your attention!


Bálint Sass.
The Verb Argument Browser.

Bálint Sass.
A unified method for extracting simple and multiword verbs with valence information and application for Hungarian.

Elena Tognini-Bonelli.
Corpus Linguistics at Work.

Tamás Váradi.
The Hungarian National Corpus.

David Yarowsky.
One sense per collocation.