Word and Multiword Expression Discovery

Timothy Baldwin and Dominic Widdows
Word Discovery
Basic Task Description

- Take an unsegmented string and dynamically insert the word boundaries:

  spotthebreaksinthisstring \rightarrow \text{spot the breaks in this string}

- Applications in:
  - the segmentation of non-segmenting languages (e.g. Japanese, Chinese, Thai)
  - segmenting up a speech signal into words
  - modelling child word acquisition
Difficulties in Word Detection

- Interdependence between word and boundary detection
- Segmentation of frequently-occurring strings
  
ad hoc, 東京都 tou·kyou·to “Tokyo prefecture”
- Inherent boundary ambiguities
  
  大学長 dai·gaku·chou “(university) president”
- Data sparseness
 Mostly-Unsupervised Statistical Segmentation of Japanese Kanji Sequences
Outline

- Unsupervised method for segmenting Japanese kanji sequences

社長兼業務部長

↓

社長 | 兼 | 業務 | 部長  sha·chou|ken|gyou·mu|bu·chou

“president and general business manager”

- Based on character n-grams (sequences of n characters)
Why the Interest in Kanji?

- Japanese consists of three basic orthographies:
  - hiragana: 46 elements (functional words, onomatopoeia, verb/adjective suffixes)
  - katakana: 46 elements (words of foreign origin, scientific names)
  - kanji: 1,945 “official” elements, more in practice (content words)

- Kanji the most productive and sparse of the three
Basic Intuition

- N-grams “windows” over a kanji string can be used as predictors of word boundaries in that:
  
  1. frequent n-grams suggest the lack of a boundary
  2. infrequent n-grams suggest the presence of a boundary

`spotthebreaks` with 4-gram window:

```
spotthebreaks
sp ot the b re a k s
sp ot
p o t t
o t t h
```
Example: *spotthebreaks*

<table>
<thead>
<tr>
<th>Word</th>
<th>N-gram</th>
<th>Freq</th>
</tr>
</thead>
<tbody>
<tr>
<td>spot</td>
<td></td>
<td>119</td>
</tr>
<tr>
<td>pott</td>
<td></td>
<td>24</td>
</tr>
<tr>
<td>otth</td>
<td></td>
<td>120</td>
</tr>
<tr>
<td>tthe</td>
<td></td>
<td>6075</td>
</tr>
<tr>
<td>theb</td>
<td></td>
<td>2118</td>
</tr>
<tr>
<td>hebr</td>
<td></td>
<td>218</td>
</tr>
<tr>
<td>ebre</td>
<td></td>
<td>49</td>
</tr>
<tr>
<td>brea</td>
<td></td>
<td>328</td>
</tr>
<tr>
<td>reak</td>
<td></td>
<td>246</td>
</tr>
<tr>
<td>eaks</td>
<td></td>
<td>57</td>
</tr>
</tbody>
</table>
Method

- For each potential boundary point:
  - For each n-gram order (e.g. 4):
    - For each of the n-grams to the left and right of the boundary:
      - What is the proportion of n-grams straddling the boundary that are less frequent than the n-gram adjoining it?

\[ \nu_n = \frac{1}{2(n-1)} \sum_{d \in \{L,R\}} \sum_{j=1}^{n-1} I_{>}(\#(s^n_d), \#(t^n_j)) \]
Returning to our Example

- For the boundary spot|theb:
  \[ v_4 = \frac{1}{2(4-1)} (1 + 2) = 0.5 \]

- For the boundary pott|hebr:
  \[ v_4 = \frac{1}{2(4-1)} (0 + 1) \approx 0.17 \]
Combining N-gram Orders

- So as to balance up the effects of data sparseness, the method advocates “voting” across different n-gram orders:

\[ v_N(k) = \frac{1}{|N|} \sum_{n \in N} v_n(k) \]

- Place boundaries at locations \( l \) where EITHER:
  1. \( v_N(l) > v_N(l - 1) \) and \( v_N(l) > v_N(l + 1) \) OR
  2. \( v_N(l) > t \)

\( t = \) threshold constant
Evaluation

- Extract n-gram statistics from 150MB of newswire data (N.B. unannotated)
- Hand-annotated 5 held-out datasets containing 500 kanji sequences each
- For each held-out dataset, use 450 sequences for parameter training \((N, t)\) and the remaining 50 sequences for evaluation (hence mostly-unsupervised)
Words vs. Morphemes

- Two separate evaluations were carried out, based on:
  - **words**: a stem and its affixes are considered as a single unit
    - [電話器] \([deN\cdot wa\cdot ki]\) “telephone (device)”
  - **morphemes**: stems and affixes are each individual units
    - [電話][器] \([deN\cdot wa]/[ki]\) “[telephone][device]”
Parameter Optimisation

- Run the algorithm over each set of 450 sequences using the different values of $N$ and $t$

- Optimise according to one of:
  - word precision $= \frac{\text{correct proposed brackets}}{\text{proposed brackets}}$
  - word recall $= \frac{\text{correct proposed brackets}}{\text{actual brackets}}$
  - word F-measure/score $= \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$

- Evaluate according to the optimal parameter setting
Results (Briefly)

- Word performance better than established Japanese morphological analyzers (JUMAN, Chasen)
- Word performance considerably better than morpheme performance
- Single-character morphemes greatest source of errors
Summary

- Method proposed for segmenting up Japanese kanji sequences based on simple n-gram statistics and voting

- Basic intuition that n-grams straddling word boundaries will be less-frequent than those adjacent to word boundaries

- Impressive results achieved for such a conceptually simple algorithm
Multiword Expression Discovery
**Basic Task Description**

- Identify the multiword expression (MWE) types in tagged (or raw) text from observation of the token distribution

- Recall: *a MWE is made up of multiple words and is syntactically and/or semantically idiosyncratic*

- Considerable body of literature on *collocation extraction*
Complications in MWE Discovery

- Working out the extent of the collocation (phrase boundary detection)
  
  \[\text{trip the light} \times \]
  
  \[\text{trip the light fantastic} \checkmark \]
  
  \[\text{trip the light fantastic at} \times \]

- Fine line between collocations and simple default lexical combinations

  \[\text{buy a car/purchase power} \]
Collocations vs. MWEs

- A collocation (≈ institutionalised phrase) is an arbitrary and recurrent word combination
- Collocations tend to stand in opposition to anti-collocations = lexically-marked synonym-substitutes
- Collocations can be semantically-marked (e.g. dark horse) but tend to be compositional (e.g. strong coffee)
- Predicative collocations vs. rigid NPs vs. phrasal templates
<table>
<thead>
<tr>
<th></th>
<th>unblemished</th>
<th>spotless</th>
<th>flawless</th>
<th>immaculate</th>
<th>impeccable</th>
</tr>
</thead>
<tbody>
<tr>
<td>eye</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>gentleman</td>
<td>-</td>
<td>-</td>
<td>?</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>home</td>
<td>?</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>?</td>
</tr>
<tr>
<td>lawn</td>
<td>-</td>
<td>-</td>
<td>?</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>memory</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>?</td>
</tr>
<tr>
<td>quality</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>record</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>reputation</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>taste</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
</tr>
</tbody>
</table>

(adapted from Cruse (1986))
Retrieving Collocations from Text: Xtract
Outline

• **Automatic method** for extracting collocations from raw text based on n-gram statistics

• **Basic intuition:** collocations are more rigid syntactically and more frequent than other word combinations

• **Method:** attempt to capture this intuition using the basic statistics of word combinations
Stage 1: Extract Significant Bigrams

- \( w \) and \( w_i \) co-occur (\( w_i \) is a collocate of \( w \)) if they are found in a single sentence separated by fewer than 5 words.

- A bigram \( (w, w_i) \) is significant iff:
  - \( w \) and \( w_i \) co-occur more frequently than chance
  - \( w \) and \( w_i \) appear in a relatively rigid configuration

- Divide up the set of collocates according to POS
Example Corpus

... multiword\(_{-1}\) expressions ...
... multiword\(_{-1}\) expressions ...
... dialect\(_{-1}\) expressions ...
... dialect\(_{-2}\) and expressions ...
... expressions of interest\(_{2}\) ...
... multiword\(_{-1}\) expressions ...
... collocation extraction ...
... expressions dialect\(_{1}\) ...
... multiword\(_{-1}\) expressions ...
... multiword\(_{-1}\) expressions ...
... multiword\(_{-1}\) expressions ...
# Noun Co-occurrence Table

\( w = \text{expressions}, \ D = \{-2, -1, 1, 2\} \)

<table>
<thead>
<tr>
<th>collocate</th>
<th>multiword</th>
<th>dialect</th>
<th>collocation</th>
<th>interest</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p_{i}^{-2} )</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( p_{i}^{-1} )</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( p_{i}^{1} )</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( p_{i}^{2} )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>( f_{req_i} )</td>
<td>5</td>
<td>3</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>( w_{i} \in C )</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
</tr>
</tbody>
</table>
Statistics of Expectation

- $\bar{f} = \frac{\sum_{w_i \in C} freq_i}{|C|} = \frac{5+3+1}{3} = 3$ (frequency average)

- $\sigma = \sqrt{\frac{\sum_{w_i \in C} (freq_i - \bar{f})^2}{|C|}} = \sqrt{\frac{(5-3)^2 + (3-3)^2 + (1-3)^2}{3}} \approx 1.63$ (strength)

- $k_i = \frac{freq_i - \bar{f}}{\sigma}$

- $\bar{p}_i = \frac{\sum_{j \in D} p^j_i}{|D|}$ (pair count average)

- $U_i = \frac{\sum_{j \in D} (p^j_i - \bar{p}_i)^2}{|D|}$ (pair count variance)
Collocation Filters

- **Strength**: $k_i > k_\alpha (= 1)$
  - select frequent collocates

- **Spread**: $U_i > U_0 (= 1)$
  - select spiky distributions

- **Peakiness**: $p_i^j \geq \overline{p}_i + (k_\beta \times \sqrt{U_i})$
  - identify interesting spikes

  $k_\beta = 0.5^{1}$

---

$^{1}$Value of 10 suggested for $k_\beta$ in Smadja (1993)
Back to our Example: Strength

- $w_1$ (multiword)
  - $k_1 = \frac{5-3}{1.63} = 1.22 > 1$ ✓

- $w_2$ (dialect)
  - $k_2 = \frac{3-3}{1.63} < 1$ ❎

- $w_4$ (interest)
  - $k_3 = \frac{1-3}{1.63} < 1$ ❎
Spread and Peakiness

\( w_1 \) (multiword)

- \( \bar{p}_1 = \frac{0 + 5 + 0 + 0}{4} = 1.25 \)
- \( U_1 = \frac{(0 - 1.25)^2 + (5 - 1.25)^2 + (0 - 1.25)^2 + (0 - 1.25)^2}{4} \approx 20.31 > 1 \)

<table>
<thead>
<tr>
<th>( p_i )</th>
<th>1.25 + (0.5 \times \sqrt{20.31}) \approx 3.50</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p_1^{-2} )</td>
<td>0 &lt; 3.50 ( \times )</td>
</tr>
<tr>
<td>( p_1^{-1} )</td>
<td>5 ( \geq ) 3.50 ( \checkmark )</td>
</tr>
<tr>
<td>( p_1^{1} )</td>
<td>0 &lt; 3.50 ( \times )</td>
</tr>
<tr>
<td>( p_1^{2} )</td>
<td>0 &lt; 3.50 ( \times )</td>
</tr>
</tbody>
</table>
Stage 2: Bigrams to N-grams

- Independent filter to detect larger N-grams

- Method: for each fixed-distance collocate \((w, w_i^j)\), extract out contiguous word sequences where \(\max(p(word[i])) > T(= 0.75)\)
Example

\[ w = \text{resistance}, \ w_i^{-3} = \text{path} \]

Concordances from the BNC:

... trod the path\(_{-3}\) of least resistance , ...

... finding the path\(_{-3}\) of least resistance will ...

... along the path\(_{-3}\) of least resistance .

... the safest path\(_{-3}\) of least resistance through ...

... took the path\(_{-3}\) of least resistance and ...

\(\Rightarrow\) the path of least resistance is a rigid noun phrase
**Stage 3: Add Syntax**

- Use *cass* dependency parser to assign syntactic labels to bigrams

- Same basic thresholding filter as used in Stage 2
Summary

- Two methods proposed for extracting collocations based on distributional statistics
  - 1st method based on the intuition that collocations occur frequently in relatively fixed configurations
  - 2nd method based on the predictability of contiguous lexical contexts for given concordances

- Basic method illustrated for determining the syntax of extracted collocations
Reflections

- *(At the time)* groundbreaking research on collocation extraction
- Not effective at extracting out low-frequency words
- Difficulties in evaluating the results of collocation extraction (applies to this day)
- Difficulties in extracting non-contiguous (predicative) collocations such as verb particles
Extracting the Unextractable: A Case Study on Verb-particles
Verb-particle Constructions (VPCs)

- **VPC = verb + obligatory particle(s)** (e.g. *hand in*, *battle on*)
  - *intransitive*: e.g. *the team battled on*
  - *transitive*: e.g. *Kim handed the paper in*

- **Variable word order for transitive VPCs:**
  - *joined*: *hand in the paper*
  - *split*: *hand the paper in*
Linguistic Properties of VPCs

- **Transitive VPCs** undergo the particle alternation (*hand in the paper vs. hand the paper in*)

- **With transitive VPCs**, pronominal objects must be expressed in the split configuration (*hand it in vs. *hand in it*)

- **Manner adverbs** cannot occur between the verb and particle (*hand it promptly in*)
The Joys of VPC Extraction

- Limited coverage of linguistic tests
- Variable word order
- Variable window length
- Structural/analytical ambiguity:
  - `hand [the paper] [in] [here] vs. hand [the paper] [in here] vs. hand [the paper in here]
  - `hand [in] [the paper] vs. hand [in the paper]"
Corpus Analysis of VPCs in WSJ

- Take random sample of 200 VPCs from Alvey Tools data and randomly search for each within the WSJ
- 62/200 attested in WSJ at mean frequency of 5.1 and median frequency of 2
- Approx two-thirds of VPCs occur less than 3 times in data
Target Corpora

- Extract VPCs out of the Wall Street Journal section of the Penn Treebank based on output of tagger and/or chunker
- Use Brown corpus data in training (where appropriate)
- Use gold-standard POS & parse annotation in establishing upper bounds on different methods
A Word on Evaluation

- Use standard measures of precision, recall and F-score ($\beta = 1$)

- Precision = $\frac{freq(\text{valid extracted VPCs})}{freq(\text{extracted VPCs})}$

- Recall = $\frac{freq(\text{valid extracted VPCs})}{freq(\text{valid VPCs in corpus})}$

- F-score = $\frac{(\beta+1) \times \text{Precision} \times \text{Recall}}{(\beta \times \text{Precision}) + \text{Recall}}$
Method-1: Simple POS-based Extraction

- Identify particles using dedicated Penn POS tag (RP)

PROCEDURE:
1. tag the data using the Brill tagger and lemmatise using morph
2. for each particle, search back to the left up to 6 words to find governing verb
3. filter data according to set of 73 canonical particles
Method-1: Example

country_NN fund_NNS offer_VBP an_DT easy_JJ
way_NN to_TO get_VB a_DT taste_NN of_IN
foreign_JJ stock_NNS without_IN the_DT hard_JJ
research_NN of_IN seek_VBG out_RP individual_JJ
company_NNS __.
Method-1: Results

<table>
<thead>
<tr>
<th>Tagger</th>
<th>#correct/total extracted</th>
<th>Precision</th>
<th>Recall</th>
<th>$F_{\beta=1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brill</td>
<td>135/135</td>
<td>1.000</td>
<td>0.177</td>
<td>0.301</td>
</tr>
<tr>
<td>Penn</td>
<td>667/800</td>
<td>0.834</td>
<td>0.565</td>
<td>0.673</td>
</tr>
</tbody>
</table>

- High precision, but low recall
- F-score well below gold standard upper bound
- Brill tagger recognises only 2 particle types: *out* and *down*
Method-2: Simple Chunk-based Extraction

- Identify particles using dedicated CoNLL-2000 chunk tag (PRT)

PROCEDURE:

1. chunk parse Brill-tagged/lemmatised data using TiMBL (use Brown corpus for training data)
2. for each (canonical) particle, search back to the left up to 6 words to find governing verb
3. only allow noun, preposition and adverb chunks between verb and particle
Method-2: Example

[O “] [PP instead of] [VP buy] [NP mask] [PP for] [NP your kid] [O ,] [ADVP just] [VP cut] [PRT out] [NP the columnist] [NP ‘ picture] [O ...] [O .]
Method-2: Results

<table>
<thead>
<tr>
<th>Chunker</th>
<th>$\frac{\text{#correct}}{\text{total extracted}}$</th>
<th>Precision</th>
<th>Recall</th>
<th>$F_{\beta=1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>TiMBL</td>
<td>$\frac{695}{854}$</td>
<td>0.772</td>
<td>0.548</td>
<td>0.641</td>
</tr>
<tr>
<td>Penn</td>
<td>$\frac{651}{760}$</td>
<td>0.857</td>
<td>0.694</td>
<td>0.766</td>
</tr>
</tbody>
</table>

- Higher recall than Method-1, but still room for improvement
- Significant disparity between results over auto-chunked and gold-standard data
Method-3: Grammar-based Extraction

- Improve recall by looking also at canonical particles occurring as non-particle (PP, ADV) chunks
- Use chunk grammar to determine the syntactic relation between verbs and “particles”
- Classify instances as:
  - unambiguously intransitive/transitive VPC
  - unambiguously intransitive/transitive non-VPC
  - possible intransitive/transitive VPC
Method-3: Identifying VPCs

- Use chunk grammar to:
  - check that the chunks which occur between the verb and particle are maximally an NP and particle pre-modifier adverb chunk
  - check for a clause boundary or NP immediately after the particle/preposition/adverb chunk
  - check the clause context of the verb chunk for possible extraposition of an NP verbal complement

- Check congruity with VPC analysis
Method-3: Structural Ambiguity

\[ \text{NP we} \ [VP \text{ may ask}] \ [NP \text{ question}] \ [SBAR \text{ as}] \ [NP \text{ you}] \]
\[ \text{VP go} \ [ADVP \text{ along}] \ [O ,] \ldots \ \checkmark \]

\[ \text{NP it} \ [VP \text{ won't do}] \ [NP \text{ any good}] \ [PP \text{ for}] \]
\[ \text{NP anybody} \ [SBAR \text{ unless}] \ [NP \text{ employee}] \ [VP \text{ know}] \]
\[ \text{PP about} \ [NP \text{ it}] \ [O .] \ \times \]

\[ \text{VP nonperform} \ [NP \text{ loan}] \ [VP \text{ will make}] \ [PP \text{ up}] \]
\[ \text{NP only about 0.5 \%} \ [PP \text{ of}] \ [NP \text{ the combine bank}] \]
\[ \text{NP 's total loan} \ [ADJ \text{ outstanding}] \ldots \ ??? \]
Method-3: Attachment Disambiguation

For cases of structural ambiguity, attempt to resolve using log likelihood ratio (verb–particle \((VP)\), verb–NP\(_1\) head \((VN_1)\), NP\(_1\) head–particle \((N_1P)\) and particle–NP\(_2\) head \((PN_2)\):

\[
[VP \text{ hand}] [NP_1 \text{ the paper}] [PP \text{ in}] [NP_2 \text{ here}]
\]

VPC realised iff:

\[
VP \times VN_1 > N_1P \times PN_2
\]
\[
VP \times VN_1 > VP \times PN_2
\]
Method-3: Feature Abstraction

- Features describing frequency of positive/negative diagnostics for each (intrans/trans) VPC type:
  \[
  \text{INTRANS}_+ \quad \text{INTRANS}_- \quad \text{INTRANS}_{ATT} \quad \text{TRANS}_+ \quad \text{TRANS}_- \quad \text{TRANS}_{ATT}
  \]

- Combine features using rule-based approach (RULE) or TiMBL 4.1 (TIMBL), with and without the attachment disambiguation mechanism (\pm ATT)
Method-3: Results

<table>
<thead>
<tr>
<th>Method</th>
<th>#correct ( \frac{# \text{correct}}{\text{total extracted}} )</th>
<th>Precision</th>
<th>Recall</th>
<th>( F_{\beta=1} )</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>RULE-ATT</strong></td>
<td>( \frac{676}{1119} )</td>
<td>0.604</td>
<td>0.694</td>
<td>0.646</td>
</tr>
<tr>
<td><strong>TIMBL-ATT</strong></td>
<td>( \frac{615}{823} )</td>
<td>0.747</td>
<td>0.661</td>
<td>0.702</td>
</tr>
<tr>
<td><strong>PENN-ATT</strong></td>
<td>( \frac{694}{927} )</td>
<td>0.749</td>
<td>0.823</td>
<td>0.784</td>
</tr>
<tr>
<td><strong>RULE+ATT</strong></td>
<td>( \frac{951}{3126} )</td>
<td>0.304</td>
<td>0.823</td>
<td>0.444</td>
</tr>
<tr>
<td><strong>TIMBL+ATT</strong></td>
<td>( \frac{739}{1049} )</td>
<td>0.704</td>
<td>0.710</td>
<td>0.707</td>
</tr>
<tr>
<td><strong>PENN+ATT</strong></td>
<td>( \frac{750}{1079} )</td>
<td>0.695</td>
<td>0.871</td>
<td>0.773</td>
</tr>
</tbody>
</table>
Method-3: Results

- Appreciable gain in recall over Method-1 and Method-2 (greater robustness over low-frequency data)
- Including attachment-resolved data bumps up recall (at expense of precision)
- TiMBL superior to rule-based method
- Relatively small disparity between gold-standard data and auto-chunked data
## System Combination

<table>
<thead>
<tr>
<th>Method</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method-1</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>Method-2</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>Method-3</td>
<td>Medium</td>
<td>High</td>
</tr>
</tbody>
</table>

- Combine methods to consolidate on relative strengths
- **Augment feature space to estimate plausibility of VPC-**
  - hood
Extra Features (1)

- Corpus frequency of:
  - particle in corpus
  - deverbinal noun and adjective forms of the VPC in the corpus (e.g. turnaround, dried-up)

- Number of letters in the verb lemma
  cf. call/ring/phone/*telephone up

- Verb and particle lemmata
## Combination Results

<table>
<thead>
<tr>
<th>Method</th>
<th>#correct ____total extracted</th>
<th>Precision</th>
<th>Recall</th>
<th>(F_{\beta=1})</th>
</tr>
</thead>
<tbody>
<tr>
<td>Combine</td>
<td>(\frac{719}{953})</td>
<td>0.754</td>
<td>0.710</td>
<td>0.731</td>
</tr>
<tr>
<td>(M_2^*)</td>
<td>(\frac{686}{778})</td>
<td>0.882</td>
<td>0.677</td>
<td>0.766</td>
</tr>
<tr>
<td>(M_3-\text{att}^*)</td>
<td>(\frac{684}{788})</td>
<td>0.868</td>
<td>0.694</td>
<td>0.771</td>
</tr>
<tr>
<td>(M_3+\text{att}^*)</td>
<td>(\frac{871}{1020})</td>
<td>0.854</td>
<td>0.823</td>
<td>0.838</td>
</tr>
<tr>
<td>Combine*</td>
<td>(\frac{1000}{1164})</td>
<td>0.859</td>
<td>0.871</td>
<td>0.865</td>
</tr>
<tr>
<td>Combine*_{Penn}</td>
<td>(\frac{931}{1047})</td>
<td>0.889</td>
<td>0.903</td>
<td>0.896</td>
</tr>
</tbody>
</table>
Final Results

- System combination with and without extra features superior to individual system performances
- Extra features produce appreciable gains in both precision and recall
- Final F-score of 0.865 (vs. upper bound of 0.896)
Summary

- Three methods proposed to extract VPCs from unannotated corpora
- System combination over the three methods proven to be effective in consolidating their relative strengths/weaknesses
- Method demonstrated to be robust over extremely low-frequency data
Acknowledgements

- The reformulation of Stage 1 of Xtract is due largely to Darren Pearce
Bibliography
