Machine Learning for NLP

SVMs for semantic error detection

Aurélie Herbelot

2018

Centre for Mind/Brain Sciences
University of Trento

Error Detection and Correction: introduction

Error Detection and Correction (EDC)

- The aim of EDC is to help L2 (or 3, or 4 or n...) learners to acquire a new language.
- **Error detection:** identify the location of an error.
- **Error correction:** suggest a replacement that would result in a felicitous sentence.

Many of the following slides were prepared by co-author Ekaterina Kochmar. Thanks for allowing re-use!

Locus of EDC

- Traditionally, EDC has focused on grammatical errors, and errors in function words.
- In English, the most frequent prepositions are:
  of to in for on with at by from
- This forms a limited confusion set to train a system on, and allows us to do *detection* and *correction* at the same time.

Preposition EDC in English

- Typically, a set of features is chosen for grammatical EDC.
- A *classifier* is then run over the possible confusion set.

Head noun	'apple'
Number	singular
Noun type	count
Named entity?	no
WordNet category	food, plant
Prep modification?	yes, 'on'
Object of Prep?	no
Adj modification?	yes, 'juicy'
Adj grade	superlative
POS ± 3	VV, DT, JJS, IN, DT, NN

Table 1: Determiner feature set for *Pick* ***the*** *juiciest apple on the tree.*

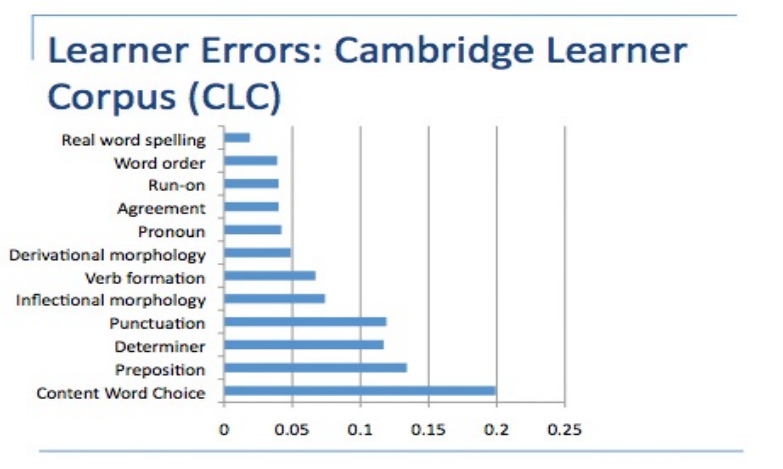
De Felice & Pulman (2008)

Lexical choice as a challenge

- Semantically related confusions:
  E.g.: *heavy decline → steep decline
  good *fate → good luck
- Form-related confusions:
  E.g.: *classic dance → classical dance
- Context-specific:
  They performed a classic Scottish dance

Errors in lexical choice (open-class / content words)

- Frequent error types [LEACOCK *et al.*, 2014; NG *et al.*, 2014]



← cover 20% of learner errors in the CLC [TETREAULT AND LEACOCK, 2014]

- notoriously hard to master
- yet, important for successful writing [LEACOCK AND CHODOROW, 2003; JOHNSON, 2000; SANTOS, 1988]

Error detection (ED) approaches

Modular

- aimed at one error type
- cast ED as a multi-class classification problem

work well with closed confusion sets and recurrent errors;

not the case with open-class words

Comprehensive

- spanning *all* error types
- example: statistical machine translation

also struggle with errors in lexical choice

Solution: Involve a semantic component

A distributional model of adjective-noun errors in learners' English (Herbelot & Kochmar 2016)

Methodology

- Focus on error detection: given a sentence, automatically detect if the chosen word combination is correct:

  They performed a ? classic Scottish dance

- Analyse content word errors from a semantic perspective ($\sim$ semantic anomaly detection in native English [Vecchi et al. (2011)])

Data

High quality annotated learner data is of paramount importance as content word errors appear to be less systematic

Learner data

[KOCHMAR & BRISCOE (2014) CLC DATASET]

- CLC: Cambridge Learner Corpus. Extracted by Cambridge Assessment from actual Cambridge exams;
- labelled with error types;
- corrections suggested;
- distinguish between stand-alone / out-of-context (*OOC*: e.g. **big inflation*) and in-context (*IC*) errors;

Example annotation

```
<AN BNCguard="0" id="1:0" lem="actual apparition_0" status="resolved" ukWac="0">
  <correction BNCguard1="5" lem1="actual appearance" ukWac1="53"/>
  <meta cand_L1="es" cand_age="21" cand_nat="AR" cand_sex="m" exam="CPE" file=
  "AR*602*8027*0300*2005*02" year="2005"/>
  <annotation>C-J-NF [= appearance]</annotation>
  <context>The role celebrities play in our society has been under discussion for a very
  long time- As a matter of fact, it's highly likely that the debate started with the
  <e t=""><c></c></e> <e t="J"><i>actual</i><c></c></e> <e t="N"><i>apparition</i><c>
  </c></e> of celebrities themselves.</context>
</AN>
<AN BNCguard="0" id="9:0" lem="ancient doctor_0" status="majority" ukWac="17">
  <correction/>
  <meta cand_L1="el" cand_age="21" cand_nat="GR" cand_sex="m" exam="CPE" file=
  "GR*802*8030*0301*2008*02" year="2008"/>
  <annotation>CO-J-N [= =]
  <comment>ADJ refers to following ADJ, not N; misparse</comment></annotation>
  <context>It is a fact that as a city has a long history that each resident can
  explain it to you and inform you about the achievements of the famous <e t="">
  <c></c></e> <e t="J"><i>ancient</i><c></c></e> Greek <e t="N"><i>doctor</i><c>
  </c></e> named "Asklipios".</context>
</AN>
```

Agreement on error annotation

- Inter-annotator agreement is given for both in-context and out-of-context ANs.
- Note: IC agreement is lower.

Annotator	1	2	3	4
1	–	0.64	0.62	0.60
2		–	0.80	0.73
3			–	0.51
4				–

Table 3: *Kappa* values, *out-of-context* annotation.

Annotator	1	2	3	4
1	–	0.52	0.44	0.48
2		–	0.56	0.36
3			–	0.43
4				–

Table 4: *Kappa* values, *in-context* annotation.

Vecchi et al (2011)

- Can compositional distributional semantics help us identify 'semantically deviant' constructions?
- **Example:** are the vectors of *hot potato* and **parliamentary potato* different?
- Investigation of different composition methods, for different features.

Vecchi et al (2011)

- **Vector neighbourhood density:** an infelicitous vector will be isolated in the space.
- **Cosine to head noun:** a *parliamentary potato* should be less a potato than a *hot potato*.
- **Vector length:** acceptable ANs should be longer than deviant ones.

Vecchi et al (2011)

	LENGTH		COSINE		DENSITY	
method	t	*sig.*	t	*sig.*	t	*sig.*
add	7.89	*	0.31		2.63	*
mult	3.16	*	-0.56		2.68	*
lm	0.16		0.55		-0.23	
alm	0.48		1.37		3.12	*

Table 1: t scores for difference between acceptable and deviant ANs with respect to 3 cues of deviance: *length* of the AN vector, *cosine* of the AN vector with the component noun vector and *density*, measured as the average cosine of an AN vector with its nearest 10 neighbours in semantic space. For all significant results, $p<0.01$.

Kochmar & Briscoe (2014)

- Can we recognised learners' errors by assuming they exhibit the same kind of deviance as the ANs studied by Vecchi et al?
- Using expanded list of features: number of close neighbours, overlap between neighbours of AN and ANs of noun/adjective, etc.
- 81% accuracy *OOC*, 65% *IC* with a decision-tree classifier.

Kochmar & Briscoe (2014)

Metric	*add*	*mult*	*alm*
VLen	0.7589	0.7690	0.1676
cosN	0.1621	**0.0248**	**0.0227**
cosA	**0.0029**	0.4782	0.0921
dens	0.6731	0.1182	0.1024
densAll	0.4967	0.1026	0.1176
RDens	0.2786	0.8754	0.1970
num	0.3132	0.4673	0.3765
OverAN	0.8529	0.1622	0.2808
OverA	**0.0151**	0.6377	0.4886
OverN	**0.0138**	0.0764	0.4118
NOverAN	0.3941	0.6730	0.0858
NOverA	**0.0009**	0.3342	0.1575
NOverN	**0.0018**	0.1463	0.1497

Table 2: p values, *out-of-context* annotation

Metric	*add*	*mult*	*alm*
VLen	0.6675	**0.0027**	**0.0111**
cosN	**0.0417**	**0.0070**	0.1845
cosA	**0.00003**	0.1791	0.1442
dens	0.4756	0.7120	0.1278
densAll	0.2262	0.7139	0.5310
RDens	0.8934	0.8664	0.1985
num	0.7077	0.7415	0.4259
OverAN	0.1962	0.8635	0.5669
OverA	**0.00007**	0.7271	0.6229
OverN	**0.0017**	0.9680	0.7733
NOverAN	**0.0227**	0.3473	0.1587
NOverA	**0.000004**	0.3749	0.1576
NOverN	**0.0001**	0.6651	0.2610

Table 3: p values, *in-context* annotation

Making sense

- Warning: humans will try to make sense of *whatever*.
- See Bell & Schäfer (2013):
  - parliamentary potato
  - sharp glue
  - blind pronunciation
- We write poetry after all...

Making sense

Dawn in New York has
four columns of mire
and a hurricane of black pigeons
splashing in the putrid waters.

Dawn in New York groans
on enormous fire escapes
searching between the angles
for spikenards of drafted anguish.

Federico García Lorca

Making sense

- See connection with notion of *lexical* sense.
- If word meaning can be shifted so drastically, how do we define lexical sense?
- Are there dictionary senses? (See Kilgarriff (1997), *I don't believe in word senses*.)

Herbelot & Kochmar (2016): overview

Focus

Errors in lexical choice within adjective-noun combinations

Contributions

1. Investigate role of context: model based on **distributional topic coherence**
2. Investigate performance across individual adjective classes: **class-dependent approach is beneficial**
3. Discuss **data size** bottleneck and challenges of **artificial error generation**

Topic coherence for error detection

Motivation

- *Topic coherence* measures semantic relatedness of words in text
- Usually applied in topic modelling [Steyvers & Griffiths (2007)]:

  E.g.: {*film, actor, cinema*} $\in$ **film** topic
- Coherence helps detect if the keywords belong together:

  E.g.: $\mathcal{COH}$({*chair, table, office, team*}) $>$ $\mathcal{COH}$({*chair, cold, elephant, crime*})

Topic coherence

Definition [NEWMAN ET AL. (2010)]

$\mathcal{COH}$ of a set of words $w_1 \ldots w_n$ is the mean of their pairwise similarities:

$$\mathcal{COH}(w_{1\ldots n}) = mean\{Sim(w_i, w_j), ij \in 1\ldots n, i < j\}$$

where $Sim(w_i, w_j)$ is estimated as the cosine distance between w_i and w_j in a distributional space

Topic coherence for error detection

Example

It was very difficult for my $\boxed{\textit{friends}}$ *to* $\boxed{\textit{call}}$ *me with the* $\underline{\textit{classical}}$ $\boxed{\textit{phone}}$

classical $\in$ **arts topic**

$Sim(classical, \{dance, music, style, literature, ...\})$ is high

In the sentence above

$$Sim(classical, \{friends, call, phone\}) < Sim(friends, call\}) < Sim(call, phone\}) < ...$$

Topic coherence system

Distributional semantics space

- Based on BNC
- 2000 most frequent lemmatised content words
- PPMI for weighting
- Context window of 10 surrounding lemmatised context words

Topic coherence estimation

- *W* – word window of *n* words surrounding the adjective-noun combination (AN)
- Measures:
  1. topic coherence $\mathcal{COH}$ of the context *W*
  2. $\mathcal{COH}_{-adj}$ of the context *W* without *adjective*
  3. $\mathcal{COH}_{-noun}$ of the context *W* without *noun*

Further implementation details

- Binary classification (correct vs. incorrect)
- *SVM* classifier through SVM^{light} [JOACHIMS (1999)] with RBF kernel
- 5-fold cross-validation experiments
- Baseline 45 to 55% with incorrect as majority

- Simple system relies on 3 $\mathcal{COH}$ features
- Extension: encode adjective as an additional feature
- Experiment with different context size *n* for *W*

Parameter choices

- Why RBF?
- C value was tuned in the range 10-200, but without significant differences in the results.

Results

	Acc ($\mathcal{COH}$)	**Acc (+*adj*)**	P_c	P_i	R_c	R_i
COH	0.59($\pm$0.03)	0.66($\pm$0.06)	0.66	0.65	0.65	0.66
K&B	-	0.65	0.62	0.72	0.69	0.58

Discussion

- Best performance for the context window of 2 words
- Performance on a par (in terms of accuracy) with the previously reported best system [KOCHMAR & BRISCOE (2014)] but the system is **much** simpler
- **More** stable in terms of P and R on both classes
- Note: adjective feature is really important.

Further analysis

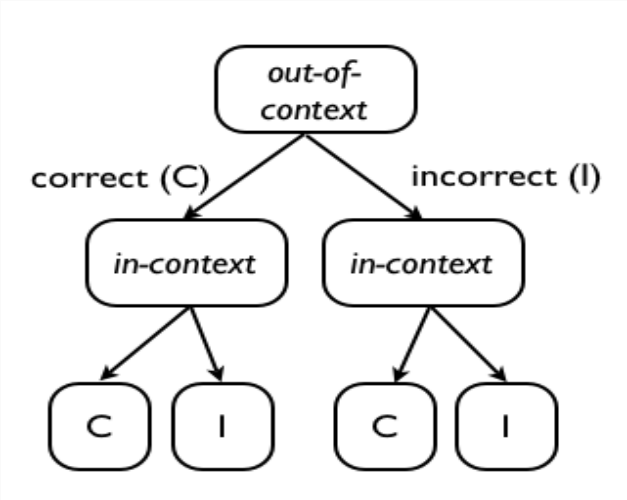
Context windows

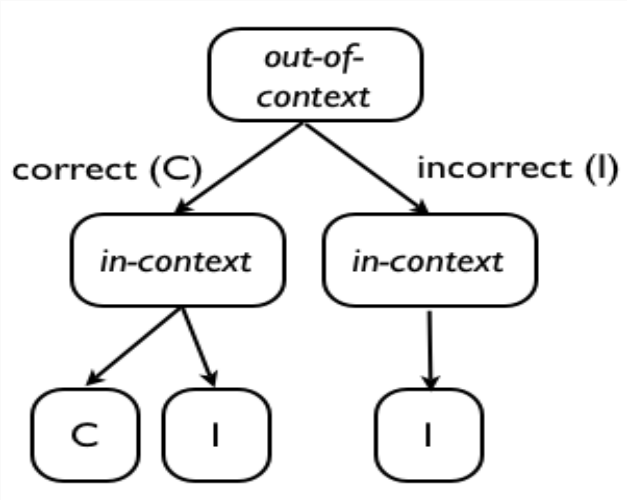
- Best results for context window $n = 2$, but the difference between different n is not statistically significant
- Hypothesis: optimal n depends on a particular instance:
  - Wider context **may harm**:
    I went shopping yesterday, and I've bought a new shirt. I had to buy it because it had a funny cat on it. It was quite cheap, it costs just £4.
  - Wider context **may help**:
    *In the second one you can eat some easy food as salads, but you also can drink a great number of *different bears .*

System combination

Out-of-context error detection

- Previous work [KOCHMAR & BRISCOE (2014)] detects errors *out of context* (*OOC*) with $Acc = 0.81$
- An ED system benefits from being aware that a combination is incorrect in general (*OOC*): if **big quantity* is incorrect **in general**, it is incorrect in **any context**





System design

Component systems

- *Context-insensitive*: COMPDIST system by [KOCHMAR & BRISCOE (2014)], uses set of compositional distributional semantics features
- *Context-sensitive*: COH system presented here

Architecture

- Concatenate features from COMPDIST and COH systems – **direct feature combination**
- Apply COH system to the output of the COMPDIST system – **pipeline**

Direct combination system: Results

System	Acc
COH	0.66
+COMPDIST	0.68

Discussion

- Features: $\mathcal{COH}$ features + *adj* + *cosine similarity* to the noun + *semantic neighbourhood* features
- Absolute improvement of 0.02 in accuracy for both *DT* and *SVM* classifiers.
- However, not statistically significant

Pipeline system

Question 1: What if we knew the true *out-of-context* label?

System	Acc
Label	0.73
+COH	0.76

Discussion

- The baseline over the *true OOC* label is very high: 73% accuracy.
- Statistically significant improvement of 0.03 in accuracy
- Shows that the COH system is useful in contextualising
- However, the gold standard label is not available in practice

Pipeline system: Results

Question 2: What is the realistic performance?

System	Acc
COMPDIST	0.64
+COH	0.67

Discussion

- The COH system is still useful in contextualising
- The difference in performance is due to lower recall on errors from the COMPDIST system

Trade-off precision/recall in real life

- The importance given to either precision or recall depends on the task.
- For EDC, it is vital that the system be precise.
- Rationale: wrongly correcting a language learner is *much worse* than not correcting them.

Class-dependent systems

System performance analysis

- Results improve with the addition of the *adj* feature
- Accuracy on the form-related errors (*classic* vs *classical*, *elder* vs *older*, etc.) is 0.77
- Performance is dependent on the adjective classified

Per-adjective precision

1 usual (4/4)	.83 strong (15/18)	.63 fast (7/11)	.41 historical (5/12)
1 rapid (1/1)	.83 clear (5/6)	.62 small (15/24)	.40 economic (2/5)
1 magic (1/1)	.80 actual (4/5)	.62 nice (39/62)	.33 deep (1/3)
1 incorrect (3/3)	.75 bad (24/32)	.62 important (18/29)	.30 whole (3/10)
1 elder (16/16)	.72 good (39/54)	.60 unique (6/10)	.25 heavy (2/8)
1 economical (33/33)	.71 hard (5/7)	.60 high (3/5)	.20 true (1/5)
1 classical (5/5)	.70 main (7/10)	.60 electric (3/5)	.18 certain (2/11)
1 classic (3/3)	.70 different (29/41)	.60 correct (3/5)	.14 precious (1/7)
.90 funny (10/11)	.69 best (36/52)	.57 near (4/7)	.14 particular (1/7)
.89 suitable (8/9)	.68 typical (11/16)	.53 wrong (7/13)	.14 ancient (1/7)
.89 soft (8/9)	.67 big (63/94)	.50 short (6/12)	0 far (0/2)
.89 full (8/9)	.66 various (6/9)	.50 present (2/4)	0 false (0/2)
.89 convenient (8/9)	.64 proper (9/14)	.47 common (8/17)	0 electrical (0/3)
.87 large (7/8)	.63 great (26/41)	.42 appropriate (3/7)	

Analysis

- Form-related confusions towards the top of the list: e.g., *economical* and *elder* yield 100%
- Adjectives expressing sentiment towards the top of the list: e.g., *funny*, *bad*, *good*, *nice*, etc.
- Wide range of precision values (25% to 87%) for quantity adjectives: e.g., *big*, *large*, *small*, *high*, etc.
- Conclusion: Different adjectives might attract different types of errors $\rightarrow$ a single classifier might not be able to model all cases

Modelling AN data: approach

- Our hypothesis: Certain adjectives might behave similarly with respect to the topic coherence $\rightarrow$ form a joint category
- However, such categories are not readily available (confusion sets for open-class words) $\rightarrow$ form categories in the empirical way
- Approach:
  1. Train 26 adjective-specific classifiers
  2. Apply to the data with **other** adjectives
  3. Record which classifier(s) perform best on each adjective
  4. The best performing classifier(s) suggest similarity between the adjectives wrt. this task

(Some) results

Adjective	Best training elements	Accuracy
appropriate	{nice, good, best, different, bad, short, fast}	71.43%
bad	{unique}	78.12%
best	{nice, good, different, fast, funny, unique}	71.70%
big	{proper}	68.09%
correct	{nice, good, best, different, bad, short, fast, unique}	80.00%
economic	{strong, typical, elder, certain}	80.00%
economical	{small, strong, typical, elder, proper, certain}	100.00%
elder	{economical, small, strong, typical, proper, certain}	100.00%
funny	{big}	90.91%
good	{nice, best, different, fast}	70.91%
great	{wrong, main}	69.05%
nice	{good, best, different, fast}	67.74%
precious	{funny}	71.43%
small	{big, proper, funny}	68.00%

Discussion

Observations

- Overall accuracy averaged over the adjectives is 0.75, which is **on a par** with human performance (0.74)
- Training on specific adjectives rather than all is **beneficial**
- Adjectives of judgement (*appropriate*, *bad*, *correct*, etc.) are best trained by other judgement adjectives
- Adjectives for form-related errors are best accounted for by the same set of classifiers
- Data size bottleneck: not enough for development phase

Ensemble-based approach

Motivation

The COMPDIST and COH classifiers also behave differently on different adjectives

COH

Best results on *large*, *bad*, *good*

COMPDIST

Better results on *short*, *heavy*

Hypothesis

Classifiers are complementary and adjective-specific combination will improve the overall result

Results

Use an oracle system that is aware of individual per-adjective classifier performance

	COMPDIST	COH	combined	oracle
Acc	0.64	0.66	0.68	**0.71**

Discussion

- Application of different classifiers **improves** the results
- Performance is **close** to human performance (0.74)
- Data size bottleneck: not enough for development phase

Error generation

Motivation

Complementary observations

- Data quality is of paramount importance
- Data size prevents the use of a separate development set

Getting more data

- Annotation is expensive and time-consuming
- Viable alternative: generate more data automatically similar to [Foster & Andersen (2009); Rozovskaya & Roth (2010)]

Approach

1. Extract examples for each adjective from the *ukWaC* corpus
2. Use 2-word context window around the AN:
   `[word_-2] [word_-1] [ADJ] [noun] [word_+1] [word_+2]`
3. Randomly shuffle the adjectives and their contexts: replace an adjective a_k in context W_k with a_m – an adjective from another context
4. Concatenate correct uses with the generated “incorrect" ones
5. Increase in the data size: $\sim$ 50% of the adjectives have a training set with $>$ 1000 instances, 93% have $\geq$ 100 training examples

Results

- Accuracy falls to 56%
- Conclusion: actual learner errors demonstrate subtle semantic phenomena that cannot be easily reproduced
- Error generation for this type of errors should be more semantically informed

What have we learnt?

- Coherence is useful.
- Performance comes at the cost of complexity.
- We cannot truly explain results.
- What does this mean?