Spelling Correction in Context

Guillaume Pinot and Chantal Enguehard

Laboratoire d'Informatique de Nantes Atlantique (LINA) — Université de Nantes — France

Introduction

The goal is to correct real-word errors. They occur when one or more modifications of a word transform it into another word which is present in the dictionary.

example : *This chocolate cake is a famous desert.*

The omission of an *s* in *dessert* reveals the word *desert*.

Intuitive idea

Traning: Learn contextual probabilities.

Correction :

1. Get all words that can have the current context.
2. Keep the words similar to the word to correct.
3. Score and sort these propositions.
4. Determine if the word is erroneous or not

Training

Let $k = 3$ and $c = 6$

We	can	have	a	lot	of	money.		
w_1	w_2	w_3	w_4	w_5	w_6		w_7	

w_3, w_4 and w_5 are in the context of w_6, that implies the sets $\{w_3, w_6\}$, $\{w_4, w_6\}$ and $\{w_5, w_6\}$.

The probability is calculated as follows: $P(w_i|w_j) = \dfrac{c_{i,j}}{c_j}$

Example of collected data:

w_i	never	before	world	of	still	and	all	I	...
w_j	before	never	of	world	and	still	I	all	...
$P(w_i\|w_j)$	0.0606	0.2352	0.0062	0.6	0.0217	0.552	0.0322	0.206	...

Similarity Between Two Words

let $\text{edist}(w_i, w_j)$ the minimal edition distance.
Let ϵ be the empty string, we can define $\text{sim}(w_i, w_j)$:

$$\text{sim}(w_i, w_j) = \begin{cases} \text{true if } \text{edist}(w_i, w_j) \leq \frac{\text{edist}(w_i,\epsilon)+\text{edist}(w_j,\epsilon)}{\gamma} + c \\ \text{false else} \end{cases}$$

In practice, we will take $\gamma = 8$ and $c = 1$. These values have been determined after several experimentations.

Detection and Correction

Let $k = 3$ and $w_c = w_5 =$ game

And	so	my	mind	game	round	to	the	business.		
w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8		w_9	

We here have w_2, w_3, w_4, w_6, w_7 and w_8 in the context of w_5, that gives

$$K_6 = \{w_2, w_3, w_4, w_6, w_7, w_8\}$$

And	so	my	mind	game	round	to	the	business.
	0.03086	0.01207		came	0.07317	0.01620	0.01571	
				same			0.00523	
				gate		0.00324	0.00305	
		0.00966		gave		0.00324	0.00174	
	0.00617			name			0.00087	
				game			0.00043	

We now need a heuristic to give a score to each proposition in order to have them in a pertinent order.

$$H_{\text{came}} = 5 + 0.030 \times 0.012 \times 0.073 \times 0.016 \times 0.0157 = 5.000000006601$$
$$H_{\text{game}} = 1 + 0.0004 = 1.0004$$

Then, the detection:

$$H1 : \ H_{\text{came}} > H_{\text{game}} \Rightarrow \text{error}$$
$$H2 : \ \lfloor H_{\text{came}} \rfloor > \lfloor H_{\text{game}} \rfloor \Rightarrow \text{error}$$

Experimentations

Corpus: *les Misérables* by Victor Hugo divided into two parts:

- The training part (480,588 words);
- The part to be corrected (53,405 words) in which errors have been added.

Adding Errors:

pire
pie pirée paire poire pitre pires spire

Results

Err.	Det.	Precision	Recall	Prop. Cor.	PCA	NPA
10%	H1	0.1081926	0.9363030	0.9622054	2.57	14.49
10%	H2	0.1561469	0.8800168	0.9635141	2.63	14.63
1%	H1	0.0206164	0.9615384	0.9500000	2.29	14.16
1%	H2	0.0301319	0.9120603	0.9696969	2.56	14.56

PCA: Position of the Correction in Average NPA: Number of Propositions in Average

Conclusion

The advantages of this algorithm are:

- simplicity;
- independence from any linguistic information;
- use of a raw corpus for the training;
- few parameters have to be regulated.

The disadvantages are:

- the significant size of the data generated by the training.
- the low precision on detection: the algorithm proposes corrections for a lot of correct words.

Futur works:

- To search better heuristics for the detection.
- To evaluate the influence of the size of the training corpus.
- To define ordered contexts to use the syntax in addition to semantics.