Detecting Learner Errors in the Choice of Content Words using Compositional Distributional Semantics

Ekaterina Kochmar
Ted Briscoe
Computer Laboratory, University of Cambridge
Cambridge ALTA Institute

University of Warwick, October 2015

Detecting **Learner Errors** in the Choice of Content Words using Compositional Distributional Semantics

- What are **learner errors** and the focus of this research

Detecting **Learner Errors** in the **Choice of Content Words** using Compositional Distributional Semantics

- What are learner errors and the focus of this research

- What are **content words** and challenges related to them

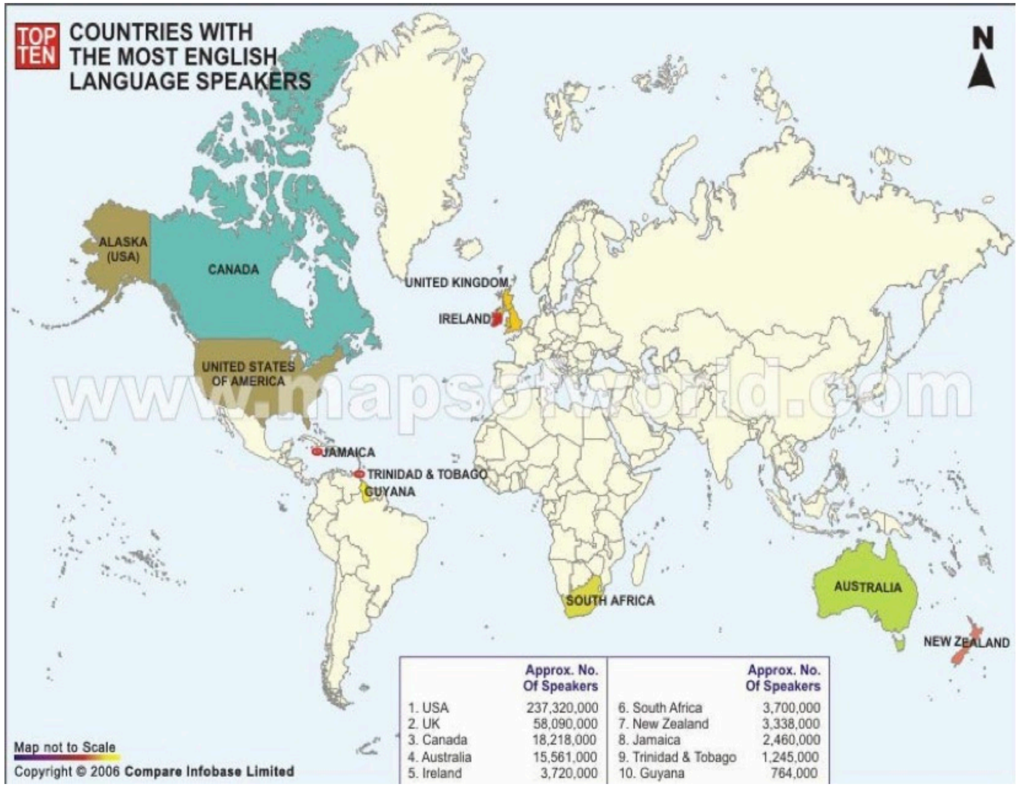
Detecting Learner Errors in the Choice of Content Words using **Compositional Distributional Semantics**

- What are learner errors and the focus of this research
- What are content words and challenges related to them
- What is **compositional distributional semantics** and how its methods are used

Detecting Learner Errors in the Choice of Content Words using Compositional Distributional Semantics

- What are learner errors and the focus of this research
- What are content words and challenges related to them
- What is compositional distributional semantics and how its methods are used
- How can a system for **error detection (and correction)** be implemented

I. Learner Errors
English Today



- About **7,000** known living languages
- Native speakers of English – about **5.52%**
- The rest – non-native speakers (language learners)
- The University of Cambridge: 18,000 students, of which **3,500** are international students from **>120** different countries

I. Learner Errors
Why this matters

Keywords: Text classification, hierarchical classification, feature selection, feature weighting

Abstract. In recent years, there have been extensive studies and <u>rapid progresses</u> in automatic text classification, which is one of the hotspots and key techniques in the information retrieval and data mining field. Feature extraction and classification algorithm are the crucial technologies for this problem. This paper firstly proposed feature extraction algorithm based on key words, the algorithm selected key words set from special part of scientific papers, and employed mutual information to extract features. And then, proposed an improved hierarchical classification method, and realized hierarchical classification of Chinese scientific papers.

Introduction

Goal of automatic text classification system is an orderly organization of the text sets, to organize the similar and related texts together. As a tool of knowledge organization, it provides more effective search strategies and more accurate query results for information retrieval.[1]

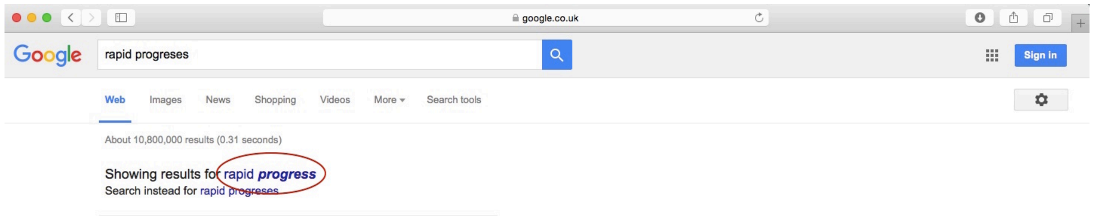
✦ In scientific text, it is particularly important that the ideas are clearly expressed

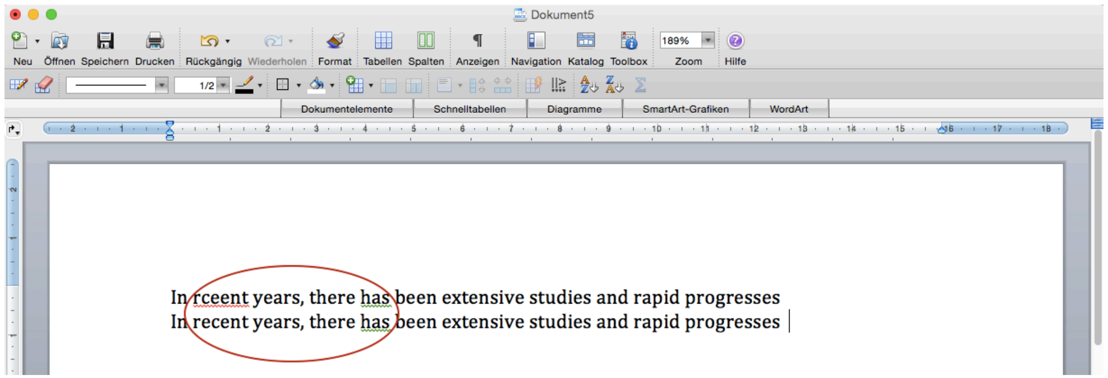
✦ What we aim to do:

- analyse the text
- detect the problematic areas
- suggest corrections
- ideally, do all of the above <u>automatically</u>

I. Learner Errors

State-of-the-art





- Currently, widely used spell-checkers and grammar-checkers can only detect and correct a **limited set** of errors (e.g., spelling, typos, some grammar)

- However, if you've picked a completely incorrect word they are unlikely to ask you if you have "***meant powerful computer instead of strong computer?***" But more on this later in the talk

I. Learner Errors
Issues

Does incorrect word choice impede understanding?

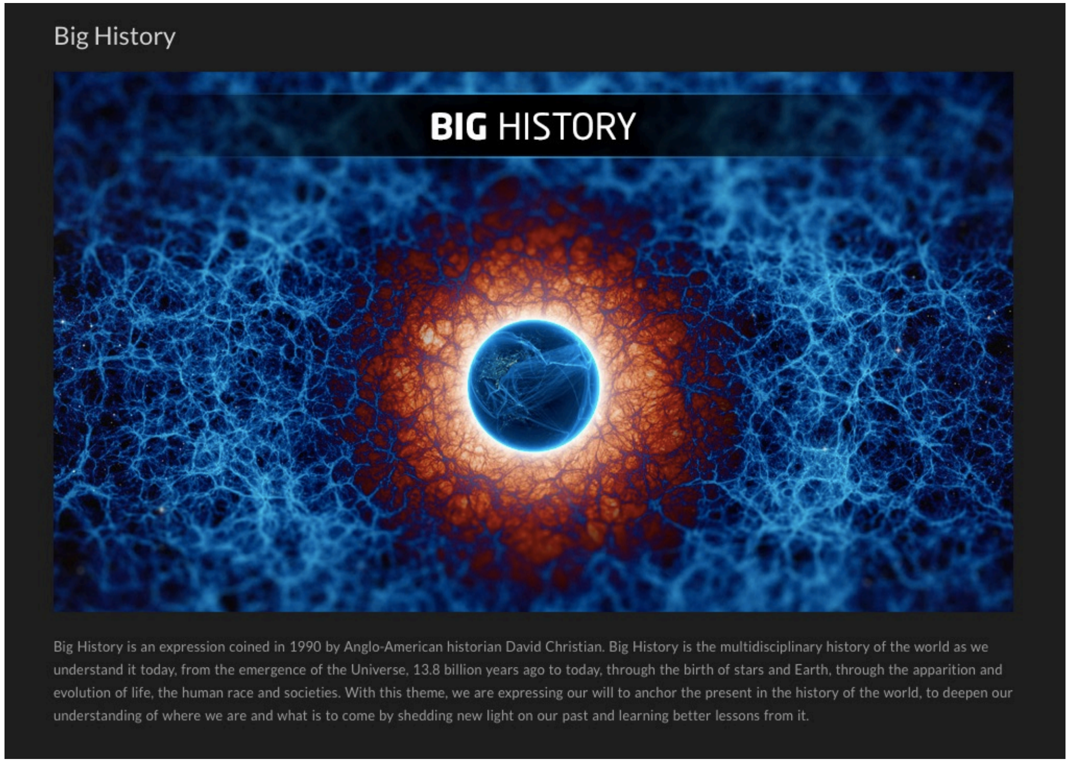
Error	Correction	Error type	Problematic to understand?
I am * student	I am **a** student	*Missing article*	
Last year I went ***in** London on a business trip	Last year I went **to** London on a business trip	*Wrong preposition chosen*	
***big** history ***large** knowledge …	**long** history **broad** knowledge …	*Wrong adjective chosen*	

I. Learner Errors
Issues

Does incorrect word choice impede understanding?

Error	Correction	Error type	Problematic to understand?
I am * student	I am **a** student	*Missing article*	?
Last year I went ***in** London on a business trip	Last year I went **to** London on a business trip	*Wrong preposition chosen*	?
***big** history ***large** knowledge …	**long** history **broad** knowledge …	*Wrong adjective chosen*	✓

I. Learner Errors
Example



Depending on the word type, the change in the original meaning can be **significant**:

When somebody uses an expression ***big history*** do they mean "*academic discipline which examines history from the Big Bang to the present*"?

I. Learner Errors
Proposed Approach

✦ Use **Natural Language Processing (NLP)** techniques:

- analyse the text
- identify the potential issues

✦ Use **Machine Learning (ML)** algorithms:

- people often use similar constructions and make same mistakes → we can learn from previous experience
- use learner data and extract error–correction patterns
- apply machine learning classifier that can learn from these patterns and can recognise them in any new text

II. Content Words

Content words vs. Function words

A bit of linguistics...

Function words	Content words
✦ link and relate the words to each other ✦ are very frequent in language ✦ examples – articles and prepositions: *I am a student at the University of Warwick*	✦ express the meaning of the expression ✦ are conceptual units ✦ examples – nouns, verbs and adjectives: *I study Computer Science at the University of Warwick. The course is very intensive*

II. Content Words
Error detection and correction for function words

- Growing interest in the field of error detection and correction in non-native texts in the recent years

- **But** most research is focusing on function words (articles and prepositions):

  - they are most frequent in language and also frequent source of errors → even if a system corrects only these types of errors it is already doing a good job

  - they are recurrent and follow repeating error–correction patterns → a lot can be learned from the data

  - they are represented with closed classes (4 articles and 10 prepositions covering 80% of all preposition uses in language) → makes error detection and correction (EDC) very suitable for machine learning classifiers

II. Content Words

EDC for function words as a machine learning problem

Example: *I am * student*

- Represent this task as a 4-class classification problem: {∅*, a, an, the*}
- Learn from the previously seen examples what the most probable correct article (***class***) is given the context of "am" and "student"
    - the contexts can be used to extract the ***features***; since errors are highly recurrent, we'll be seeing similar contexts again and again, which guarantees that we are learning something reliably from the data
    - we can even step one level up and generalise from *student* to *occupation*
    - if the classifier suggests choosing a different article in this context, detect an error and correct to the suggested one

II. Content Words

Does that mean the task is solved for content words, too?

- Errors in content words (**nouns**, **verbs**, **adjectives**) are more diverse → we cannot represent them as a general and limited number of classes and reliably learn the probabilities from the data

- The contexts are also more diverse → we might never see exactly the same context around content words again and learn anything about the features

- Corrections cannot be represented as a finite set applicable to all nouns, all verbs or all adjectives in language, and they always depend on the original incorrect word

- Content words are not just linking other words, they express **meaning** → we should take this into account

II. Content Words

Types of errors in content words

- Words are confused because they are **similar in meaning**:

Now I felt a big anger (great anger)

- Words are confused because they have **similar form**:

It includes articles over ancient Greek sightseeings as the Alcropolis or other famous places (ancient sites)

- There are some other, **less obvious** reasons:

Deep regards, John Smith (kind regards)

- Interpretation depends on the **context**, and the chosen words simply don't fit:

The company had great turnover, which was noticable in this market (high turnover)

II. Content Words
Data

- Data quality is important when it comes to machine learning approaches – we want to learn reliably from the data

- We use the ***Cambridge Learner Corpus*** (***CLC***) which is a large corpus of texts produced by English language learners sitting Cambridge Assessment's examinations (http://www.cambridgeenglish.org)

- In addition, we have collected a ***dataset of errors in content words*** that illustrate typical content word confusions (http://ilexir.co.uk/applications/adjective-noun-dataset/)

- The dataset is annotated with respect to the correctness of the words chosen and the most probable reasons for the errors (related via meaning, form or unrelated)

II. Content Words

Dataset

- The dataset contains annotation, corrections and examples extracted from the ***real*** learner data
- Stored in an XML format to facilitate the use and extraction of relevant information

http://ilexir.co.uk/applications/adjective-noun-dataset/

II. Content Words

More on the dataset

- Dataset contains 798 examples of adjective–noun (AN) combinations and 800 examples of verb–noun (VN) combinations

- 100 examples for each subset were extracted and annotated by 4 annotators to ensure reliability. We measure:

  - ***Cohen's kappa*** – measures inter-rater agreement taking into account agreement by chance p_e → is considered to be more robust

$$\kappa = \frac{p_o - p_e}{1 - p_e} = 1 - \frac{1 - p_o}{1 - p_e},$$

  - where $\boldsymbol{p_o}$ denotes ***observed (percentage) agreement***:

p_o = *(#matching annotations)/(total)*

II. Content Words

Adjective–noun (AN) dataset annotation

Annotation	Out-of-context	In-context
Agreement (p_o)	**0.8650** ± 0.0340	**0.7467** ± 0.0221
Kappa (κ)	**0.6500** ± 0.0930 (*substantial*)	**0.4917** ± 0.0463 (*moderate*)
Annotated as correct	78.89%	50.84%
Annotated as incorrect	21.11%	49.16%

II. Content Words

Verb–noun (VN) dataset annotation

Annotation	Out-of-context	In-context
Agreement (p_o)	**0.8217** ± 0.0279	**0.8467** ± 0.0377
Kappa (κ)	**0.6372** ± 0.0585 (*substantial*)	**0.6810** ± 0.0751 (*substantial*)
Annotated as correct	55.57%	39.14%
Annotated as incorrect	44.43%	60.86%

III. Semantic Approach
Overview

- ✦ We know that for content words, many errors stem from semantic mismatch – the resulting combination with the incorrectly chosen words **changes** the original meaning or **distorts** it completely
- ✦ We need to build a **computational model of the word meaning** so that a machine can understand the words and detect the anomalies
- ✦ Luckily, there are the models of **compositional distributional semantics** that can help us:
  - **distributional semantics** helps capturing individual words' meaning
  - **compositional semantics** helps successfully (or unsuccessfully) combine the individual meanings into the meaning of a longer phrase

III. Semantic Approach

Distributional Semantic Models (DSMs)

- **Key assumption**: word meaning can be approximated by a word's distribution

"You shall know a word by the company it keeps" (Firth)

- **Method**: represent words with distributional vectors, dimensions = co-occurrence with a predefined set of context words

- **Hypothesis**: semantically similar words occur in similar contexts and, therefore, will be represented with a similar vectors in the semantic space

- A nice property of a direct interpretation of word meaning through vectors in space

III. Semantic Approach
DSM example

- Try representing a meaning of word ***rose*** computationally
- ***Step 1***: collect examples of the use of the input words (e.g., ***rose***) in contexts:

  [...]

  This rose grows up to six feet tall
  The desert rose blooms in the garden
  I bought some roses and lilies the other week for just £2.50

  [...]

- ***Step 2***: use the context words and the input words to create a semantic space – a matrix that would encode the number of co-occurrences of the input and context words
- ***Step 3***: fill in the matrix with the number of co-occurrences

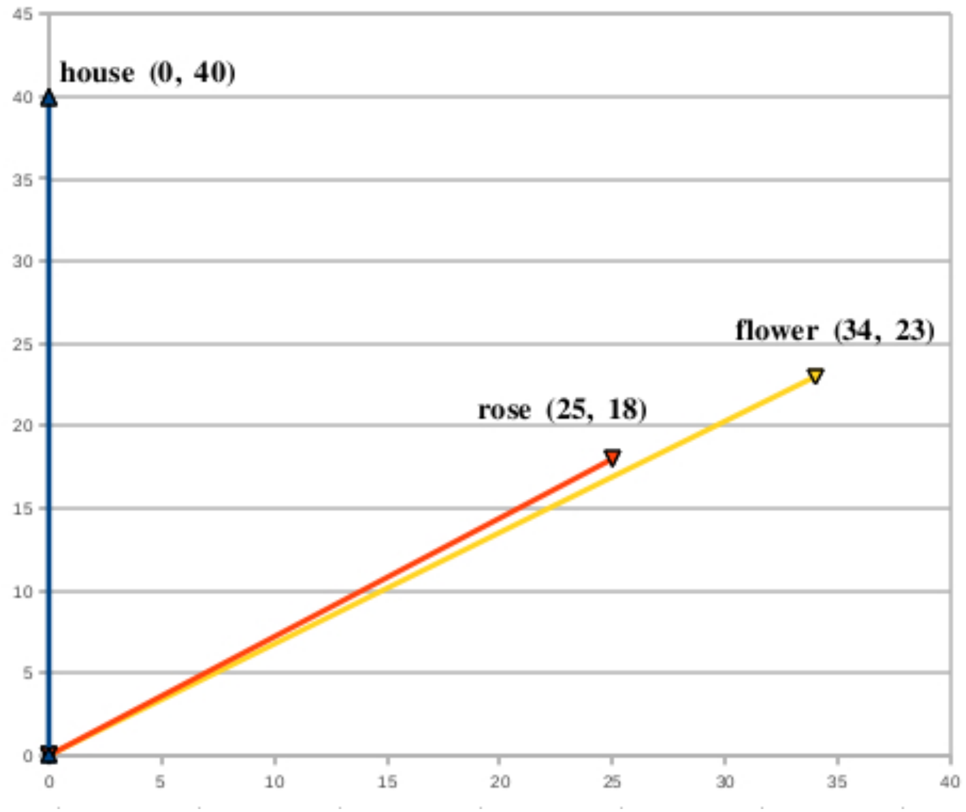
III. Semantic Approach

Semantic Space construction

	bloom	*buy*	*garden*	*grow*	*tall*	...
rose	*25*	*18*	20	33	8	...
flower	*34*	*23*	30	38	10	...
house	*0*	*40*	24	5	21	...

III. Semantic Approach

Semantic Space graphical interpretation



- We can conclude that ***bloom***, ***garden*** and ***grow*** are all characteristic of ***rose***
- One can ***buy houses*** as well as ***roses*** and ***flowers***, so this is typical for all three of them
- However, ***roses*** and ***flowers*** will in general share more properties – we can see the vectors closer together

III. Semantic Approach

Can any language expression be modeled this way?

What happens when we try applying same models to longer expressions?

- Well, we might find **100** examples with the word ***rose***, **50** of which will be about ***red roses***, **30** about ***white roses*** and **none** about ***blue roses***

- That means, longer expressions (***red rose***, ***white rose***) will necessarily have sparser and less reliable vectors

- Also, we won't be able to say anything about ***blue rose*** – if we don't see it in the data, does the object itself not exist at all? Have we just not looked carefully enough?

III. Semantic Approach

Compositional Semantics methods

Instead of relying on distributional information for longer phrases, let's use distributions of words within phrases and build vectors for longer phrases in a compositional way

- **Component-wise additive** model:

$$c_i = a_i + b_i$$

$$(\text{blue\_rose})_i = \text{blue}_i + \text{rose}_i$$

- **Component-wise multiplicative** model:

$$c_i = a_i \times b_i$$

$$(\text{blue\_rose})_i = \text{blue}_i \times \text{rose}_i$$

III. Semantic Approach

Measures of semantic anomaly

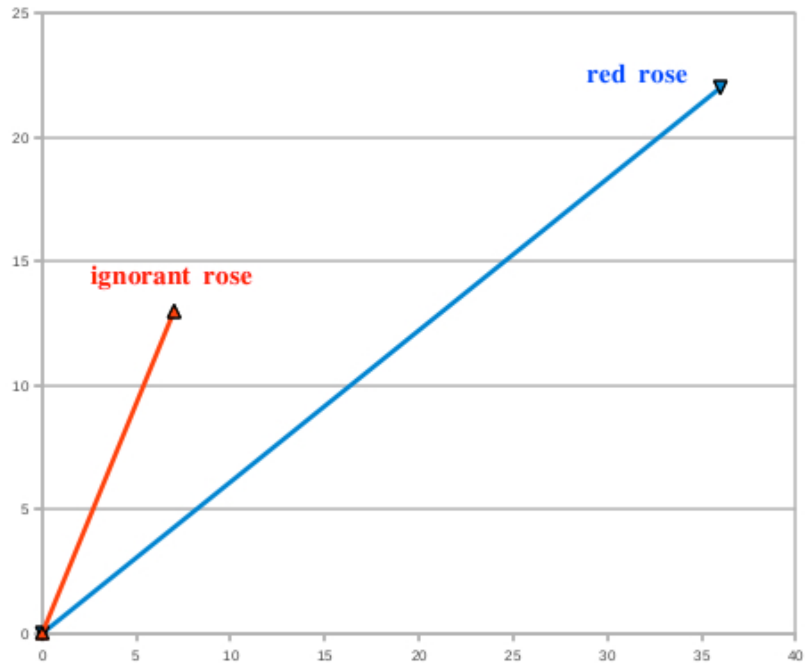
- Earlier, we have assumed that the computational semantic representation of words will tell us something about correctness of our examples
- Now, we have modeled the phrases computationally. How can we distinguish between the representations for the correct and for the incorrect phrases?

- Since there is a direct geometric interpretation for the semantic vectors, we assume that **certain properties of the vectors** will highlight the differences

III. Semantic Approach

Vector length as a measure of semantic anomaly

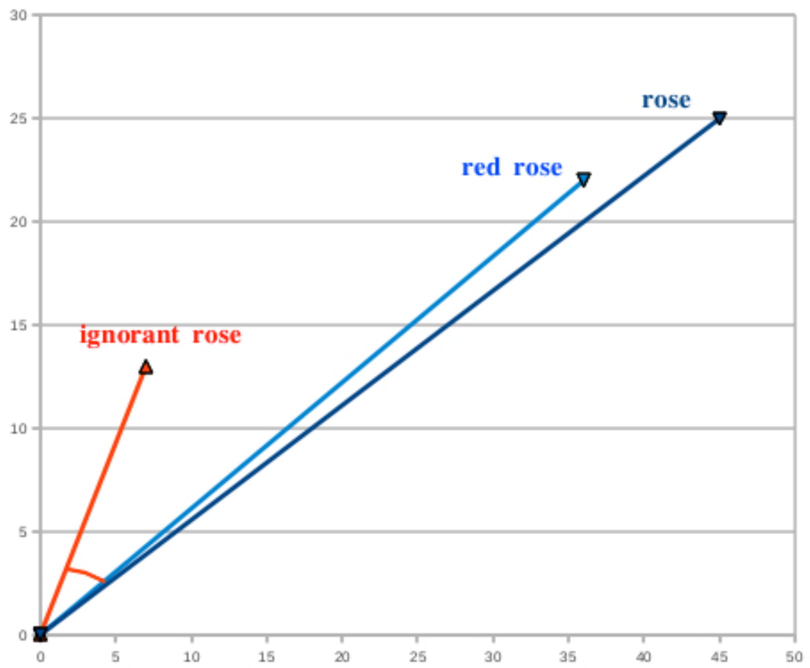
In anomalous ANs, the counts in the input vectors are distributed differently → some “incompatible dimensions” would receive low counts → anomalous AN vectors are expected to be **shorter** than vectors of the acceptable ANs



III. Semantic Approach

Cosine to the input noun as a measure of semantic anomaly

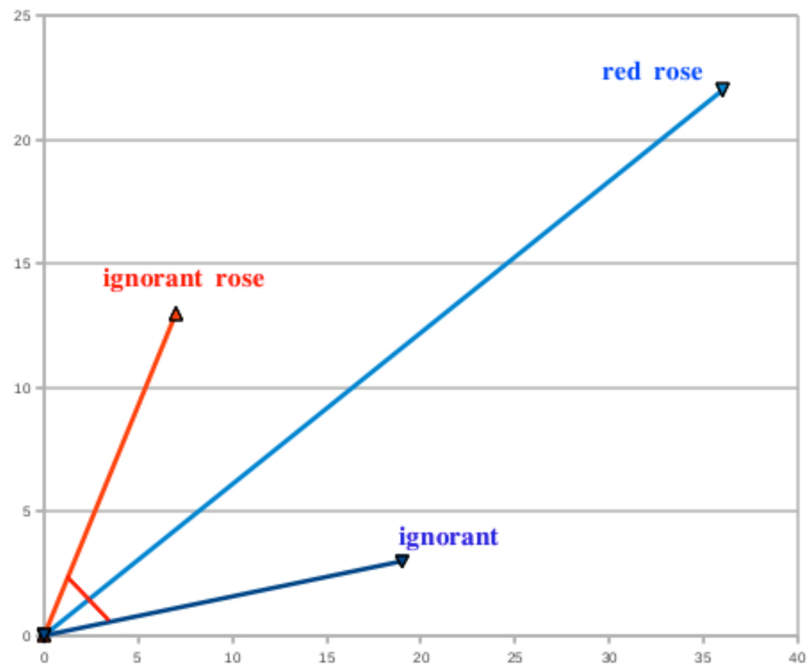
Anomalous ANs are less similar to the input nouns, and the semantic space provides a direct interpretation of the similarity of two words via their distance in the space → vectors of the anomalous ANs are expected to have **lower cosine** to the input noun vector



III. Semantic Approach

Cosine to the input adjective as a measure of semantic anomaly

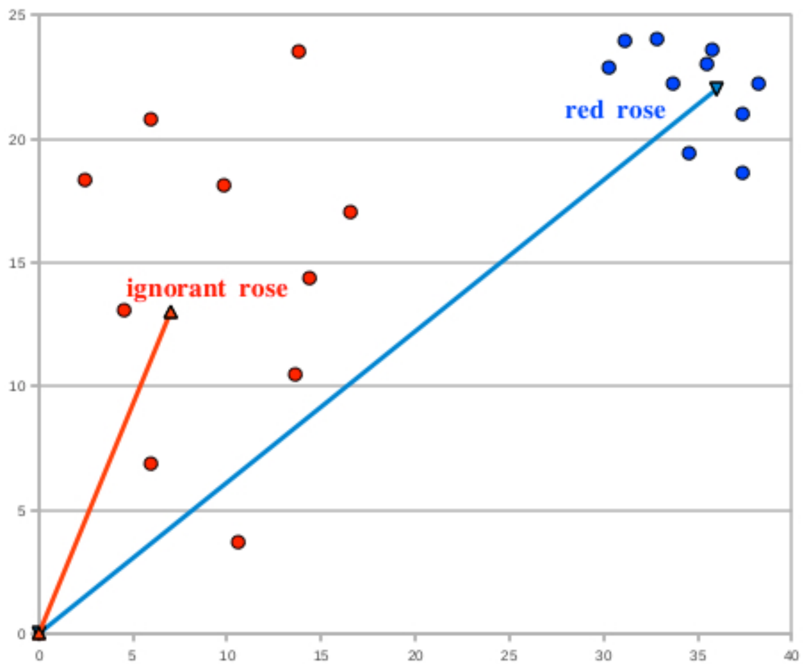
Similarly, we assume that the same holds for the input adjective: in anomalous ANs, the input adjective will be located further away in the semantic space and have a **lower cosine** with the AN than in semantically acceptable ANs



III. Semantic Approach

Neighbourhood density as a measure of semantic anomaly

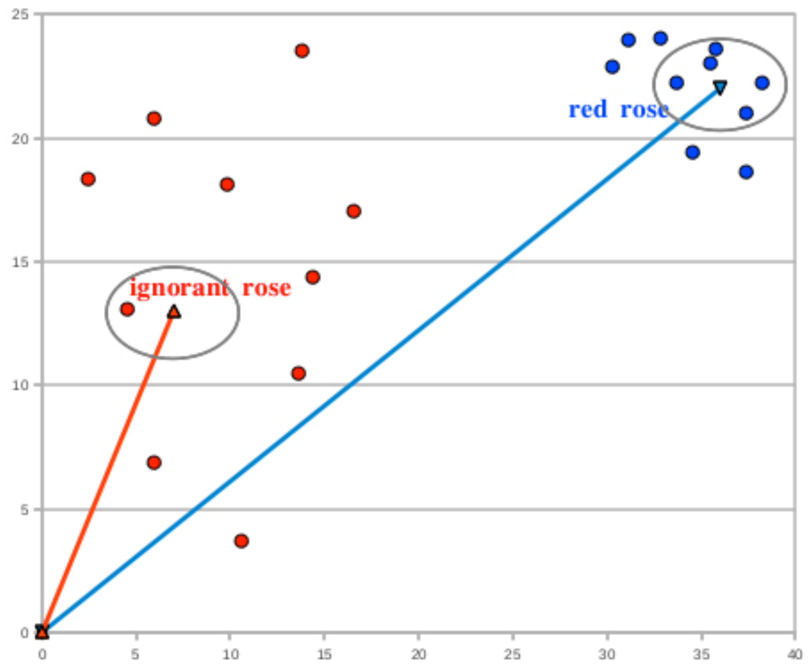
Anomalous AN vectors are expected to not have any specific meaning → they are expected to not be closely surrounded by other words with similar meaning → have sparser neighbourhoods in the semantic space. We measure this as an **average cosine** (= distance) to the 10 nearest neighbours



III. Semantic Approach

Ranked neighbourhood density within close proximity as a measure of semantic anomaly

To further explore the space of the neighbours (i.e., semantically similar words) we define ***close proximity*** as a subspace populated by vectors for which the cosine is **>0.8**, and measure **RDens** as a sum for all close neighbours ***i*** of $rank_i \times distance_i$



III. Semantic Approach

Component overlap as a measure of semantic anomaly

We assume semantically acceptable ANs to be placed in the neighbourhoods populated by **similar words and combinations**, and calculate the proportion of neighbours containing the same words as the input phrases. We expect this **proportion** to be lower for the anomalous ANs (**lower overlap**)

red rose	***ignorant rose***
• [x] **rose**	• people
• **red** [x]	• blind people
• flower	• like-minded
• ...	• ...

Ekaterina Kochmar and Ted Briscoe (2013). *Capturing Anomalies in the Choice of Content Words in Compositional Distributional Semantic Space*. In Proceedings of RANLP 2013

III. Semantic Approach

All of the above as measures of semantic anomaly

- Finally, we also need to make sure that our **hypothesis** holds and the semantic metrics actually can be used to distinguish correct phrases from the incorrect ones

- **Method**: apply ***t-test*** to check if the measures return statistically different values for the two groups of vectors – for the correct and for the incorrect phrases

Measure	**p value < 0.05***
VLen	**0.0033***
CosN	**0.0017***
CosA	**0.00002***
Dens	0.3531
RDens	**0.0002***
COver	**0.0041***

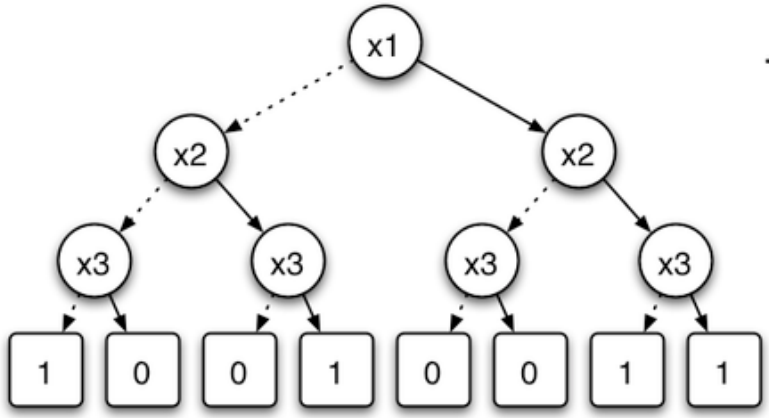
IV. ED System

Error Detection (ED) in content words as an ML task

- So far, we have seen that
  - ML approaches are widely applied to ED in function words where it is represented as a multi-class classification problem: several classes with one denoting the correct choice
  - The same approach is hard to apply to content words, yet it would be good to explore the potential of ML approaches
- We know how to capture the relevant properties of phrases to distinguish between correct and incorrect phrases
- **Solution**:
  - Cast ED in content words as a ***binary classification problem*** {correct, incorrect}
  - Use semantic properties to generate numeric ***features***

IV. ED System

Decision Tree classifier for ED



x1	x2	x3	*f*
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

- We apply ***Decision Tree Classifier*** to our classification problem
- Two classes – ***correct*** (0) and ***incorrect*** (1)
- At each node, the classifier checks whether the value of the feature falls within a certain value interval (e.g., whether ***VLen<0.5*** or ***VLen>=0.5***) and follows the relevant path
- The algorithm makes sure the most discriminative rules are applied first

IV. ED System

Decision Tree classifier algorithm

Data: data D; set of features F.

Result: feature f to split on.

$I_{min} \leftarrow 1$;

for *each* $f \in F$ **do**

  split D into subsets $D_1, ..., D_n$ according to the values v_i of f;

  if $Imp(\{D_1, ..., D_n\}) < I_{min}$ **then**

    $I_{min} \leftarrow Imp(\{D_1, ..., D_n\})$;

    $f_{best} \leftarrow f$;

  end

end

return f_{best}

Algorithm 1: `BestSplit-Class`(D, F) – find the best split for a decision tree

IV. ED System

Results

Dataset, annotation	Accuracy (averaged over 5 folds)	Lower bound (=majority class distribution)	Upper bound (=annotator agreement)
ANs, out-of-context	**0.8113** ± 0.0149	0.7889	**0.8650** ± 0.0340
ANs, in-context	**0.6535** ± 0.0189	0.5084	**0.7467** ± 0.0221
VNs, out-of-context	**0.6577** ± 0.0166	0.5557	**0.8217** ± 0.0279
VNs, in-context	**0.6491** ± 0.0188	0.6086	**0.8467** ± 0.0377

Ekaterina Kochmar and Ted Briscoe (2014). *Detecting Learner Errors in the Choice of Content Words Using Compositional Distributional Semantics*. In Proceedings of COLING 2014

Further evaluation of the ED system

- ***Precision*** = #(instances that belong to class ***n*** & are identified by the system as belonging to class ***n***) / #(all instances identified by the system as belonging to class ***n***)

$$\text{Precision} = \frac{tp}{tp + fp}$$

- ***Recall*** = #(instances that belong to class ***n*** & are identified by the system as belonging to class ***n***) / #(instances in the data that belong to class ***n***)

$$\text{Recall} = \frac{tp}{tp + fn}$$

- ***F-measure*** – harmonic mean of the two

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

	Predicted (+)	Predicted (-)
Actual (+)	***tp***	***fn***
Actual (-)	***fp***	***tn***

IV. ED System

Class-specific performance of the ED system

Combination type	Precision	Recall	F1
ANs, out-of-context, ***correct***	0.8193	0.9762	0.8909
ANs, out-of-context, ***incorrect***	0.7500	0.2488	0.3736
ANs, in-context, ***correct***	0.6173	0.7226	0.6558
ANs, in-context, ***incorrect***	0.7071	0.5898	0.6409

Ekaterina Kochmar and Ted Briscoe (2014). *Detecting Learner Errors in the Choice of Content Words Using Compositional Distributional Semantics*. In Proceedings of COLING 2014

IV. ED System

Class-specific performance of the ED system

Combination type	Precision	Recall	F1
VNs, out-of-context, ***correct***	0.6497	0.8688	0.7434
VNs, out-of-context, ***incorrect***	0.6837	0.3767	0.4858
VNs, in-context, ***correct***	0.6027	0.3192	0.4174
VNs, in-context, ***incorrect***	0.6637	0.8630	0.7503

Ekaterina Kochmar and Ted Briscoe (2014). *Detecting Learner Errors in the Choice of Content Words Using Compositional Distributional Semantics*. In Proceedings of COLING 2014

IV. ED System

Summary on the ED system

- We have showed that our algorithm detects errors with high accuracy

- There is still some room for improvement – it is close to, but does not yet reach human performance on this task

- The features derived using semantics and trying to capture the meaning of the words are useful

- The algorithm shows high precision → it is reliable → learners can use it to detect errors in their writing

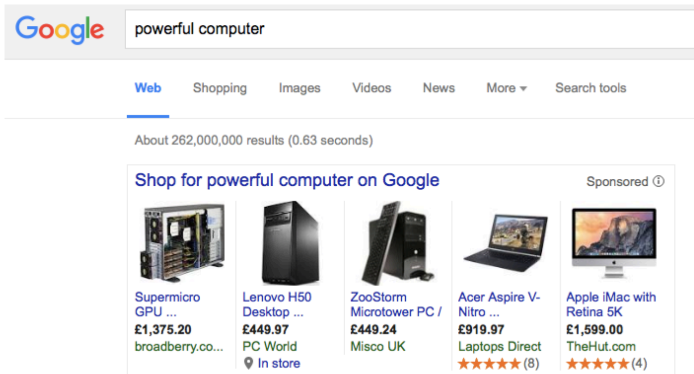
- Major source of mistakes by the algorithm – in cases where confusion occurs due to **similarity in meaning**: **small speech* vs *short speech*, **rise punctuality* vs *increase punctuality*

V. EDC System

Correction of the errors

- Once errors are identified, the learners/users will want to know how to **correct** them

- Something like "***Did you mean powerful computer instead of strong computer?***" will be helpful

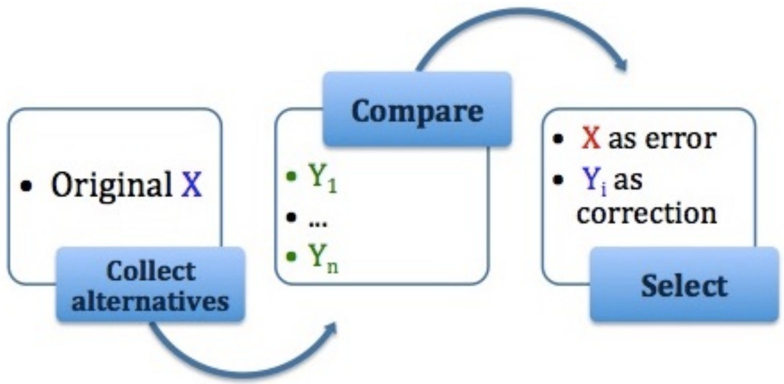


V. EDC System

How to perform error correction?

- Before, we have already noted that there is no finite set of corrections suitable for **all** nouns, or **all** adjectives, or **all** verbs – the particular set of corrections depends on the original word choice

- Once we identify an error, we need to **collect** all possible corrections, **rank** them, and **suggest** the most probable one



V. EDC System

Where to look for corrections?

- Our data exploration suggests that most frequently people confuse words
  - similar in meaning (***powerful*** ~ ***strong***)
  - similar in form (***economic*** ~ ***economical***)
  - related to their first languages (***good humor*** vs ***good mood***, from *French* **bon humor**)
- Luckily, there are resources where we can find the suggestions
  - **WordNet** – a large database where content words are organised into groups representing similar concepts
  - **Levenshtein distance** – helps to estimate how many one-letter deletions, insertions or substitutions are required to convert one string to another
  - **CLC** – information on real learner confusion patterns and their probabilities

V. EDC System

Use of different resources for error correction

What we hope to cover using different resources:

- Levenshtein distance (**Lv**): form-related error patterns:

  **electric society → electronic society*

  *important *costumer → important customer*

- WordNet (**WN**): meaning-related error patterns:

  **heavy decline → steep decline*

  *good *fate → good luck*

- **CLC**: first language-related error patterns:

  **strong noise → loud noise*

  *historical *roman → historical novel*

V. EDC System

Coverage of different resources for error correction

Measure **coverage** as the proportion of one-word corrections that can be found in different resources

Resource	Coverage
LV	0.1588
WN	0.4353
CLC	0.7912
CLC+LV	0.7971
CLC+WN	0.8558
All	**0.8618**

V. EDC System

Create alternative phrase corrections

- Using the possible corrections for adjectives and possible corrections for nouns, generate the corrections for ANs:

{alternative ANs} = ({alternative adjs} × noun) & (adjs × {alternative nouns})

- Rank the suggestions using frequency in a big corpus or a more sophisticated measure – normalised pointwise mutual information (**NPMI**)

$$NPMI(AN) = \frac{PMI(AN)}{-log_2(P(AN))}$$

$$PMI(AN) = log_2 \frac{P(AN)}{P(A)P(N)}$$

- Additionally, offset taking the typical learner error–correction pattern probabilities **CP** into account: given **M** is frequency or NPMI, estimate

$$\begin{aligned} M' &= M \times CP(a_{orig} \rightarrow a_{alt}) \\ &\quad \times CP(n_{orig} \rightarrow n_{alt}) \end{aligned}$$

V. EDC System
Error correction system assessment

✦ Mean reciprocal rank (**MRR**) showing how high in the list of proposed alternatives the appropriate correction is scored

$$MRR = \frac{1}{|N|} \sum_{i=1}^{|N|} \frac{1}{rank_i}$$

✦ The higher the rank – the better:

- **MRR=1** shows that the appropriate correction is always scored #1
- **MRR=0.5** shows that the appropriate correction is always scored #2
- **MRR=0.33** shows that the appropriate correction is always scored #3

V. EDC System

Error correction results

Resource	MRR
CLC_freq *CLC_NPMI*	**0.3806** 0.3752
(CLC+Lv)_freq *(CLC+Lv)_ NPMI*	0.3686 0.3409
(CLC+WN)_freq *(CLC+WN)_NPMI*	0.3500 0.3286
All_freq *All_NPMI*	0.3441 0.3032
All_freq' ***All_NPMI'***	**0.5061** 0.4843

Ekaterina Kochmar and Ted Briscoe (2015). *Using Learner Data to Improve Error Correction in Adjective–Noun Combinations*. In Proceedings of the 10th Workshop on Innovative Use of NLP for Building Educational Applications

V. EDC System

Break-down of the results

Top N system suggestions	% cases covered
1	**41.18**
2	49.12
3	56.77
4	61.77
5	65.29
6	66.18
7	67.35
8	68.53
9	69.71
10	**71.18**
Not found at all	25.29

Ekaterina Kochmar and Ted Briscoe (2015). *Using Learner Data to Improve Error Correction in Adjective–Noun Combinations*. In Proceedings of the 10th Workshop on Innovative Use of NLP for Building Educational Applications

Thank you!

- Further information:
  - http://www.cl.cam.ac.uk/~ek358/
  - Ekaterina.Kochmar@cl.cam.ac.uk
- Datasets:
  - http://www.cambridgeenglish.org
  - http://ilexir.co.uk/media/an-dataset.xml
  - http://ilexir.co.uk/applications/adjective-noun-dataset/