Computational Models of Problems with Writing of English as a Second Language Learners

Huichao Xue

Dissertation Director

Rebecca Hwa, PhD, Computer Science Department

Committee

Janyce Wiebe, PhD, Computer Science Department

Milos Hauskrecht, PhD, Computer Science Department

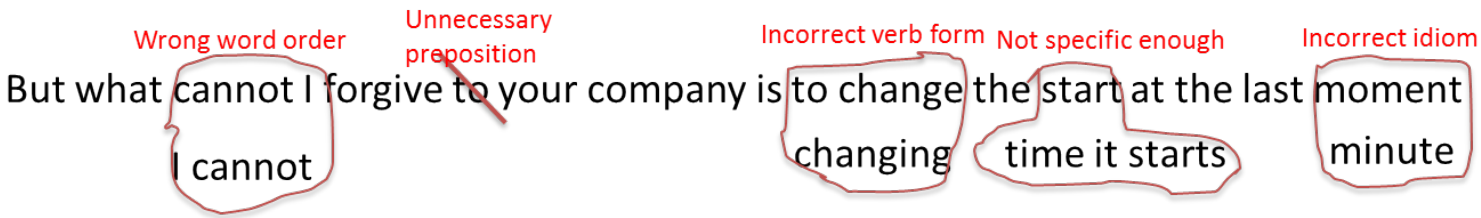
Joel Tetreault, PhD, Yahoo! Labs

Sep 26, 2014

Local Feedbacks to Language Learners

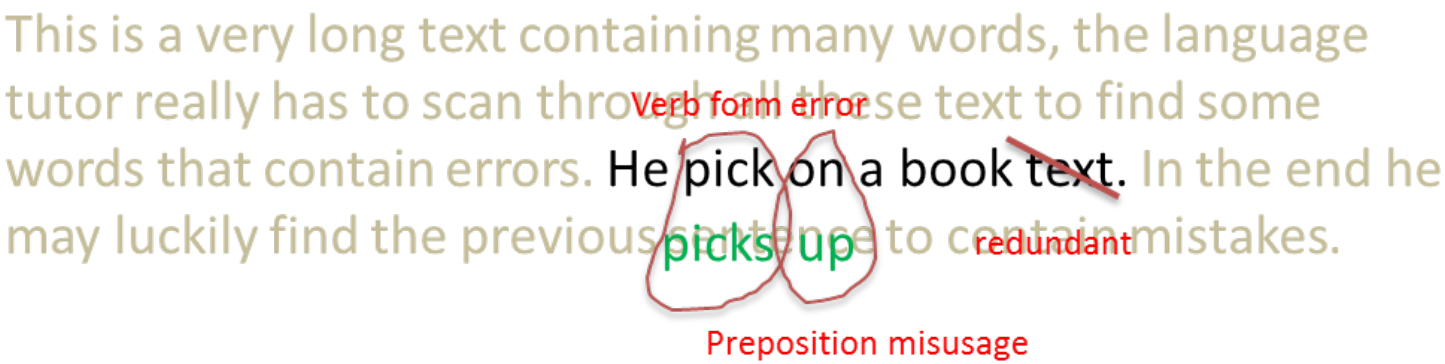
ESL learners need language tutors feedbacks.

- General feedbacks – e.g. Essay scoring (Attali & Burstein, 2006)
- Localized feedbacks – *location* of mistakes, *reason*, *how* to fix them



Providing localized feedbacks requires human effort.

Language Tutor's Efforts



Manually providing localized feedbacks

- Tutors need to
  1. Scan over the text, to identify writing mistakes
  2. Provide feedbacks in a consistent way

Fully automated feedback generation systems (Leacock *et al.*, 2010) need human knowledge

- Hiring tutors to annotate large corpora
- Consulting language tutors

Computer Programs to Assist Tutors

- Highlight suspicious areas

  This is a very long text containing many words, the language tutor really has to scan through all these text to find some words that contain errors. He pick on a book text. In the end he may luckily find the previous sentence to contain mistakes.

- Auto-complete tutors' feedbacks

- Build fully automated system components w/o supervisions

The success depends on how much computers can make smart decisions for us.

Highlighting Suspecious Areas

Current computer programs can highlight errors on closed word sets

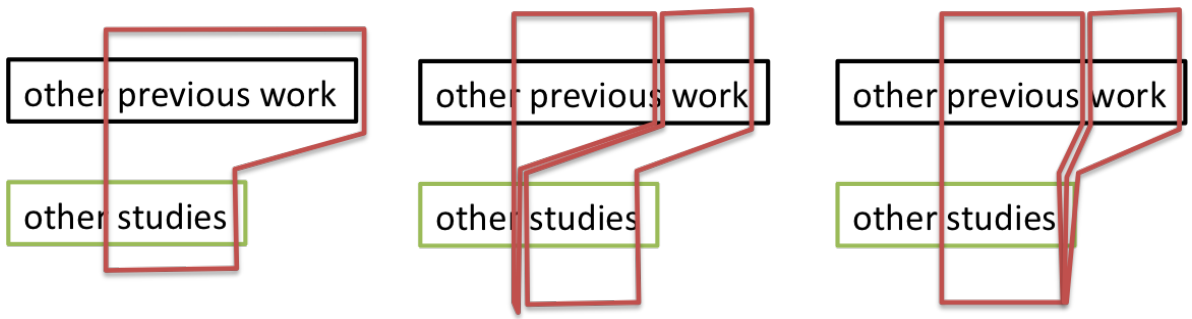
- Verb agreement errors (e.g. MS Word)
- Preposition/determiner errors (Tetreault *et al.*, 2010; Han *et al.*, 2010; Rozovskaya & Roth, 2010b)

Not handled: redundancies – the 2nd most common feedback in NUCLE (Dahlmeier *et al.*, 2013)

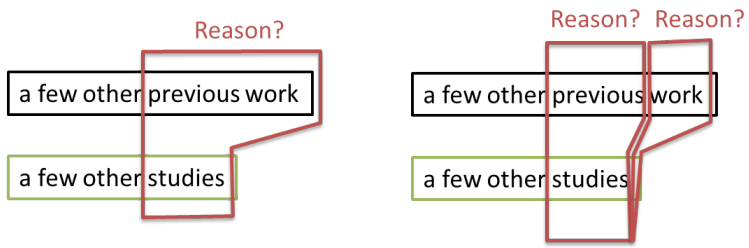
> . . . short comings in ~~the~~[Function word] technology .
> . . . the price will ~~keep~~[Open class word] continue to go higher.
> . . . what ~~are the things that~~[Multi-word phrase] governments should . . .

Auto-completing Tutors' Feedbacks

- First thing to do: correction detection – determining how many individual corrections are made (Swanson & Yamangil, 2012)
- There are often multiple ways to interpret



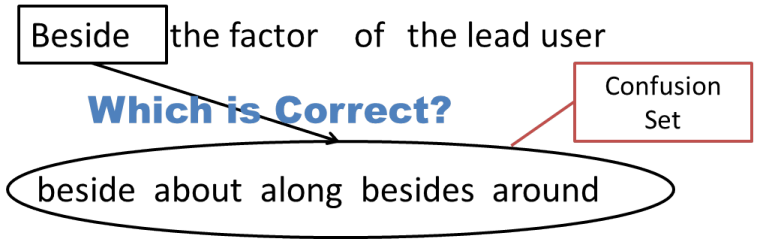
- S&Y's system's correction detection outcomes are incorrect 30% of the time.



Constructing System Components without Supervision

We consider *confusion sets*: a list of words that learners are most likely confused between.

1. GEC systems need to consult confusion sets (Tetreault *et al.*, 2010; Liu *et al.*, 2010)



2. Previously confusion sets were built with human efforts

Directly by tutors (Dahlmeier & Ng, 2011a; Liu *et al.*, 2010)

Collecting statistics from annotated corpora (Rozovskaya & Roth, 2010b)

- to →for: 35 times
- for → of: 82 times
- to → at: 1 time

.....

Computational Models

Mathematical formulas that can automate decision making.

Classification	object $\Rightarrow$ categories
Language models	sentence $\Rightarrow$ fluency score
Machine translation	words $\Rightarrow$ other words
Distance Based Models	objects $\Rightarrow$ similarity/distance between them

I build computational models to automate language tutors' tasks

- **Redundancy detection**: find the most redundant word within a sentence (Part I)
- **Correction detection**: better isolate individual corrections from tutors' revisions (Part II)
- **Confusion set construction**: infer which words are more confusable (Part III)

Thesis Claim

Computational models can reduce human effort in both providing feedbacks and building fully automated systems.

Contributions

1. The first study on automating redundancy detection.
   - 1.1 Proposed the redundant word detection task – picking the most redundant word/phrase within a given sentence.
   - 1.2 A corpus to support redundant word detection
   - 1.3 A redundant word detector with 37% accuracy
2. Proposed a model to infer individual corrections that occur in a proposed revision with 80% accuracy.
3. Building confusion sets without human supervisions.

Outline

- Introduction
- Challenges
  - Steps in applying computational models to tutors' tasks
  - Challenges in redundancy detection
  - Challenges in correction detection
  - Challenges in confusion set construction
- Part I – Redundancy detection
- Part II – Correction detection
- Part III – Confusion set construction
- Conclusions

Automating Language Tutors' tasks

- Steps in applying computational models
  1. Identifying a task that can be automated by computational models

     e.g. choose the right preposition under a context (Tetreault *et al.*, 2010)
  2. Find corpora/data that supports the task

     e.g. ESL corpora with annotations on preposition mistakes
  3. Develop a model that captures the most relevant factors

     e.g. a classifier to choose the right preposition
- Applying computational models for our proposed tasks face challenges in these steps

Annotated ESL Corpora

- Available annotated ESL corpora help us build computational models

> As technology continues to advance at an ever increasing rate, energy is also being consumed at a rapid pace. Oil is still the main source of energy that drives the economy despite its ill effects on the environment. Furthermore, it is projected to be depleted in fifty ~~years~~years'(Npos) time. Therefore, the global issue of today has to do with clean and sustainable energy. The six Generation IV nuclear reactors that are currently being researched under the supervision of the Generation IV International Forum (GIF) are the answers to this very problem. As of now, the Lead-Cooled Fast Reactor (LFR) and the Sodium-Cooled Fast Reactor (SFR) ~~are looking~~appear(Wtone) very promising and will likely ~~to~~(Vform) be the main choice of nuclear power plant in the near future. However, further comparison of both the reactors ~~base~~based(Vform) on cost, performance and safety actually revealed (LFR) as the better option.

- Summary of corpora used in this dissertation

Corpus	# of sentences
NUCLE (Dahlmeier & Ng, 2011b)	61,625
HOO2011 (Dale & Kilgarriff, 2010)	966
FCE (Yannakoudakis *et al.*, 2011)	33,900
UIUC (Rozovskaya & Roth, 2010a)	2,221

Challenges in Automating Redundancy Detection – Task 1

- Not straightforward to define the computational task – redundancy is *subjective*
  - Example: . . . is a new term for us[redundant?] . . .
- Current corpora's annotation reliability is low (e.g. NUCLE).

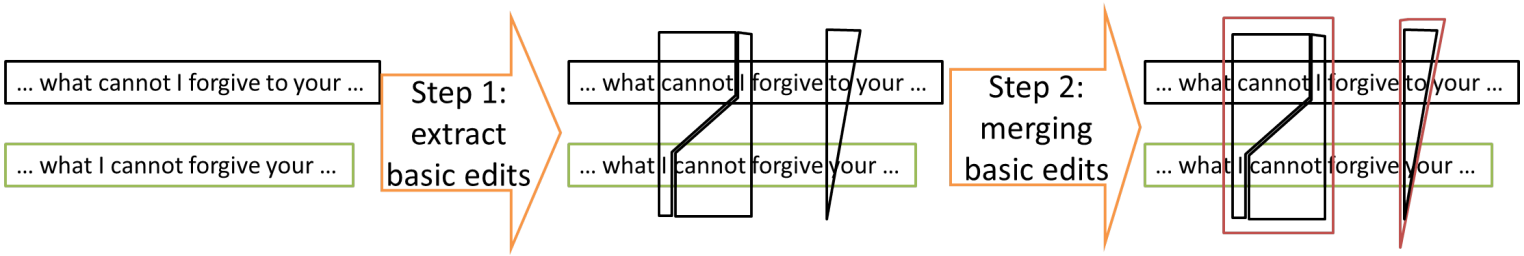
. . . prevent most accidents *to occur*. ⇒ redundant
. . . might cause accidents *to occur*. ⇒ unchanged
. . . preventing crimes *to occur*. ⇒ "from occurring"

- Measuring redundancies has not been studied before

Related work (XX)	XX but not redundant	Redundant but not XX
Grammar Error Correction (Leacock et al. 2010)	He ~~like~~ likes dogs.	… illustrate the ~~methodological~~ challenge …
Sentence compression – keep words that are specific to the sentence (Jing 2000; McDonald 2006; Clarke and Lapata 2007)	Kurtz completed ~~in high platform~~ diving.	These findings are often unpredictable ~~and uncertain~~.
Sentence simplification (Coster and Kauchak, 2011)	… positive critical reception … → … good reviews …	… not only just …

Challenges in Correction Detection – Task 2

1. Choosing the right sub-task to improve.
   - S&Y's system operates in two steps



   - Which step is the bottleneck?
2. Building a better model for that task.
   - Step 1 – the focus of previous work
     - Phrase extraction (Koehn *et al.*, 2003) and paraphrase extraction (Cohn *et al.*, 2008; Snover *et al.*, 2009; Heilman & Smith, 2010).
   - Step 2 – granularity, is less a concern in previous work

Challenges in Automatic Confusion Set Construction – Task 3

The challenge is to build a model that captures relevant factors for confusions

- Confusions are developed during word learning

- We build models to simulate word learning

Word learning concerns the interaction between

- Out-of-context meaning: what the word means, and how it relates to other words (lexical semantics)
  - e.g. *of* means "belonging to, relating to, or connected with"
- In-context usage: how to use the word (language modeling)
  - e.g. gift *of*? you

Previous models focus on one single component.

Overview of Part I – Redundancy Detection

1. I propose to automate a less subjective sub-task
   - Our annotation study suggests: automate detecting the most redundant area in a sentence
   - I propose – redundant word detection.
2. I build a redundant word detector, with 37% accuracy.
   2.1 I propose a machine learning framework for redundant word detectors
   2.2 I propose two features to capture redundancy, for one word's contribution to *meaning* and *fluency*

Sentence/Word level redundancy

It is difficult to fully automate redundancy detection – people may not agree on its outcomes

Although GM food is a new term *for us*[Redundant?], ...

We turn to automating its sub-tasks

- Does a given sentence read wordy? – sentence level redundancy
- What are the most redundant words in the sentence? – word level redundancy

We can automate the less subjective sub-task. Which one is it?

Annotation Study – Will native English speakers agree more on sentence or word level?

We collect annotations on both tasks with Amazon Mechanical Turk (USA, HIT Approval Rate $>$ 96%, Approved HITs $>$ 500 (Akkaya *et al.*, 2010))

Sentence-level (0-3)	Word-level annotations (at least one span)
3	~~But~~ in reality , burning of fossil fuels for energy are damaging the environment and public health ~~too~~ , which may ~~, to a certain extent ,~~ ~~causes~~*cause* even more damage than a safe operating nuclear power plant .
3	But in reality , burning of fossil fuels for energy are damaging the environment and public health too , which may ~~, to a certain extent ,~~ ~~causes~~*cause* even more damage than a safe operating nuclear power plant .

- On sentence level: we calculate the κ score
  - $\kappa = 0.03$
- On word level: we calculate how frequently one person's annotation overlaps/includes the other person's
  - Overlaps: 42%; Includes: 35%.

We choose to automate word-level redundancy detection.

Our task : Redundant Word Detection

- Most redundancies occur on single words

Corpus	Percentage of single word redundancies
NUCLE	67.55%
FCE	86.93%
Turker Annotations	70.42%

- Redundancy word detection – highlight the most redundant word within a sentence

> However, I believe that if GMF has a very bright future. The usage amount of chemical substances will be largely reduced. One major problem is that whether GM food will do harm to human bodies.

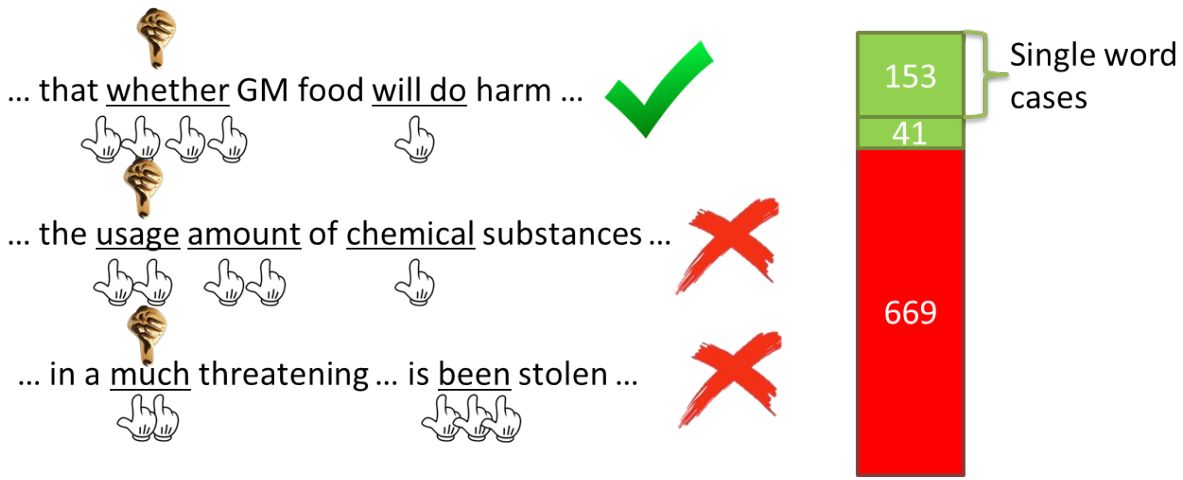
- Such a system can reduce the tutors' focus onto one word per sentence – 5% words.
  - A better than random system would help

Corpus to Support Redundant Word Detection

- Ideal corpus: sentences with one clearly most redundant word.
- Current corpora contain cases where it is hard to define the *most* redundant word

Equally Most Redundant:	. . . terrorism will result in a much threatening situation if nuclear weapon is been stolen .
OK to delete either one:	So the usage amount of chemical substances will be largely reduced .

- We currently do not consider these cases – we filter them out.

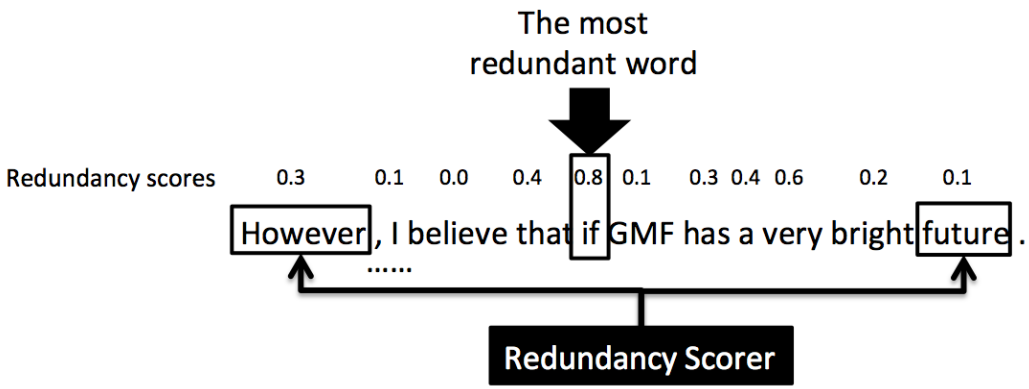


Redundancy Scorers

Research question: Can we automate redundancy detection for language tutors?

- Task: redundant word detection
- Corpus: the 153 sentences
- Model: ?

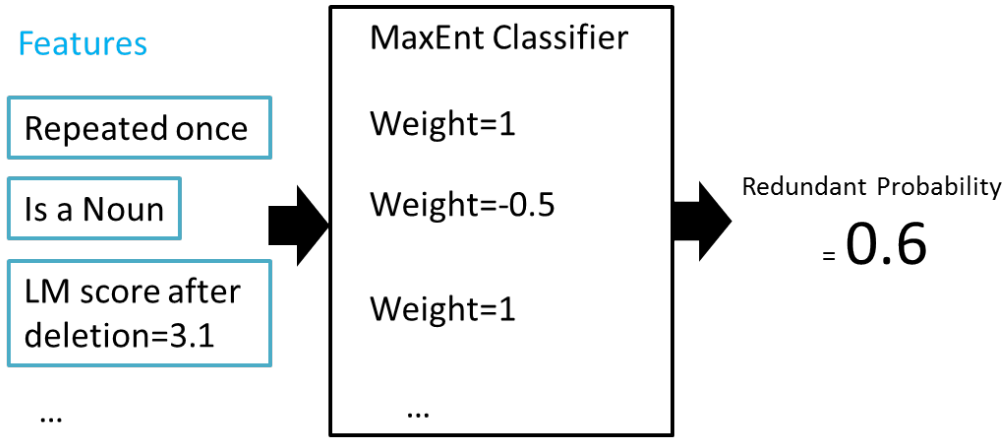
System overview



We use MaxEnt classifier to implement the scorer

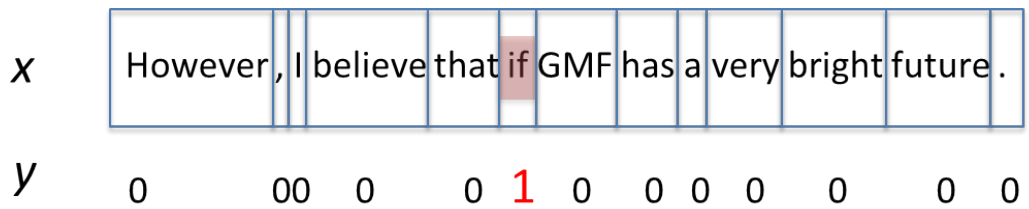
- Input: a word
- Output: Probability it is redundant $\Rightarrow$ redundancy score

Using the MaxEnt classifier (logistic regression)



We can tune the weights with our collected annotations:

1. We extract n training instances from sentence of n words

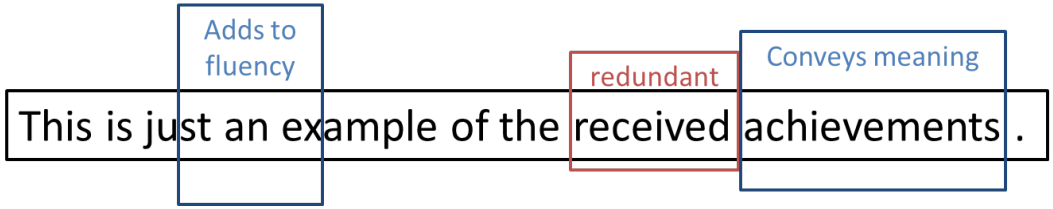


2. Run a training algorithm

Question: what *features* would capture redundancy?

Approximating Meaning with Translation/Alignment

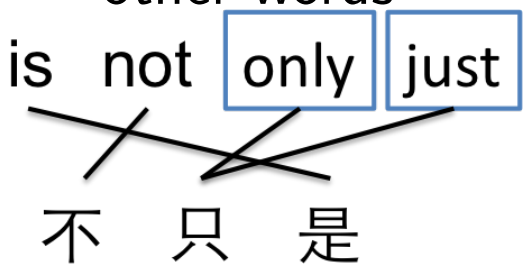
A word is *redundant* if: deleting it results in a *fluent* English sentence that conveys the same *meaning* as before



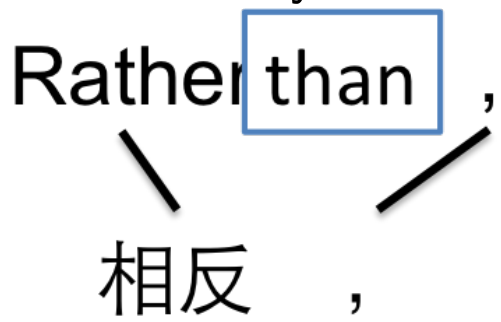
How to capture one word's contribution to meaning?

- Translation in another language (Hermet & Désilets, 2009; Madnani *et al.*, 2012).
- A word's alignment suggests how much meaning it conveys

Carrying same meaning as other words'



Not semantically meaningful



Formalizing Redundancy with Translation/Alignment

Redundancy as Translation Probability

- A word e_k in e is deemed redundant if we translate sentence e into a foreign language sentence f^* and then back into English, we are likely to obtain the rest of the sentence $e_{\_}^{k}$

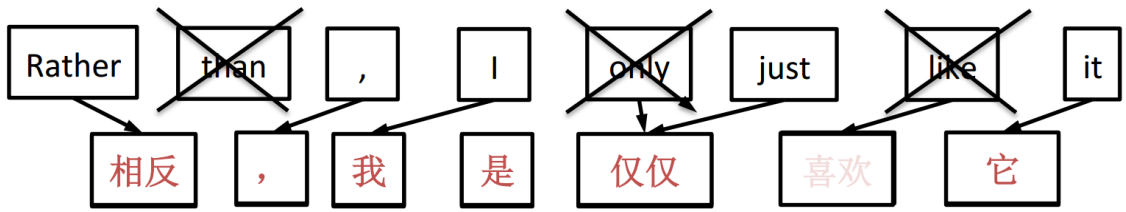
$$R(k;e) \approx \underbrace{\log \Pr(f^*|e)}_{\text{constant}} + \underbrace{\log \Pr(e_{\_}^{k}|f^*)}_{\text{Our Focus}}$$

- For example: $R(\text{“only”})$ in “I only just like it”:

$$R \approx C + \log \Pr(\underbrace{\text{“I just like it”}}_{e_{\_}^{k}} | \underbrace{\text{我是仅仅喜欢它}}_{f^*})$$

The Role of Alignments in Translation Probabilities

- We use IBM model 1 (Brown *et al.*, 1993)

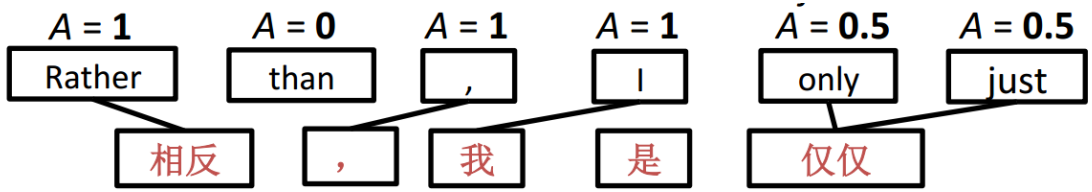


Two Features – Fluency and Meaning

(Xue and Hwa, EACL 2014)

$$R(k;e) \approx \underbrace{\mathrm{LM}(e_{-}^{k})}_{\text{fluency w/o } e_k} + \underbrace{A(k)\log \Pr(e_k)}_{\text{Meaning redundancy}} + C(e)$$

- $\mathrm{LM}(e_{-}^{k})$: log likelihood of sentence *without* e_k
  - A word is redundant, if deleting it does not hurt fluency
- Meaning redundancy
  - $A(k)$: number of words aligned with e_k

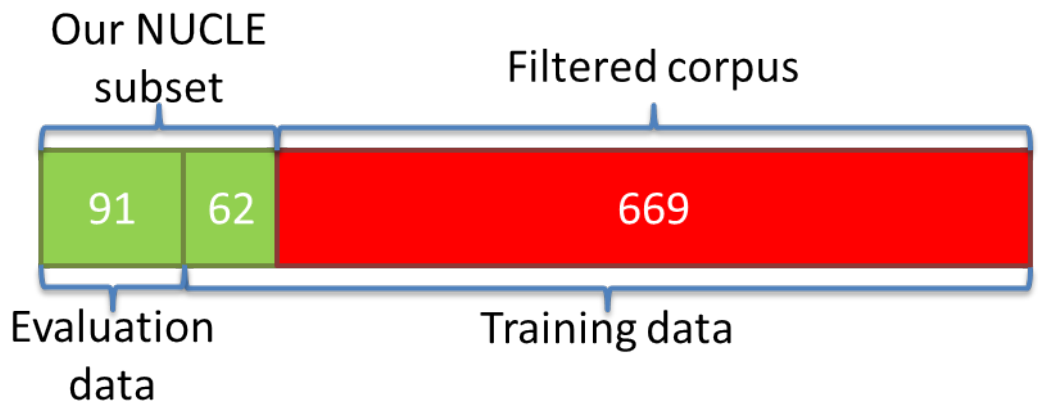


  - $\Pr(e_k)$: unigram probability of e_k
    - Rare words are often less redundant

Experiments

1. Can we automate redundant word detection?
2. What measures/features can help detect the most redundant word?
3. Is approximate meaning with translations helpful for redundancy detection?
   - 3.1 If so, what pivot language would work the best?
   - 3.2 How do our proposed measures of *meaning* and *fluency* interact?

- Dataset



- We use languages available on Google translate

Measures of Redundancy

We compare different redundancy measures by

- Training a MaxEnt classifier to incorporate features
- Using them in isolation

Redundancy Features

- Binary
  - POS, POS-bigram, unigram
  - occurs twice, beginning of sentence
  - contrast word (e.g. yet), negation word (e.g. but), connector (e.g. however), preposition (e.g. of), determiner (e.g. the)
- Numerical
  - round-trip: number of words disappeared after a round-trip translation (Madnani *et al.*, 2012)
  - LM: fluency, by trigram language model
  - sig-score: sentence compression (Clarke & Lapata, 2007)
  - Contrib: meaning preservation $A(k) \log \Pr(e_k)$
  - LM+Contrib: the proposed feature.

Different Feature Combinations (Fr as pivot)

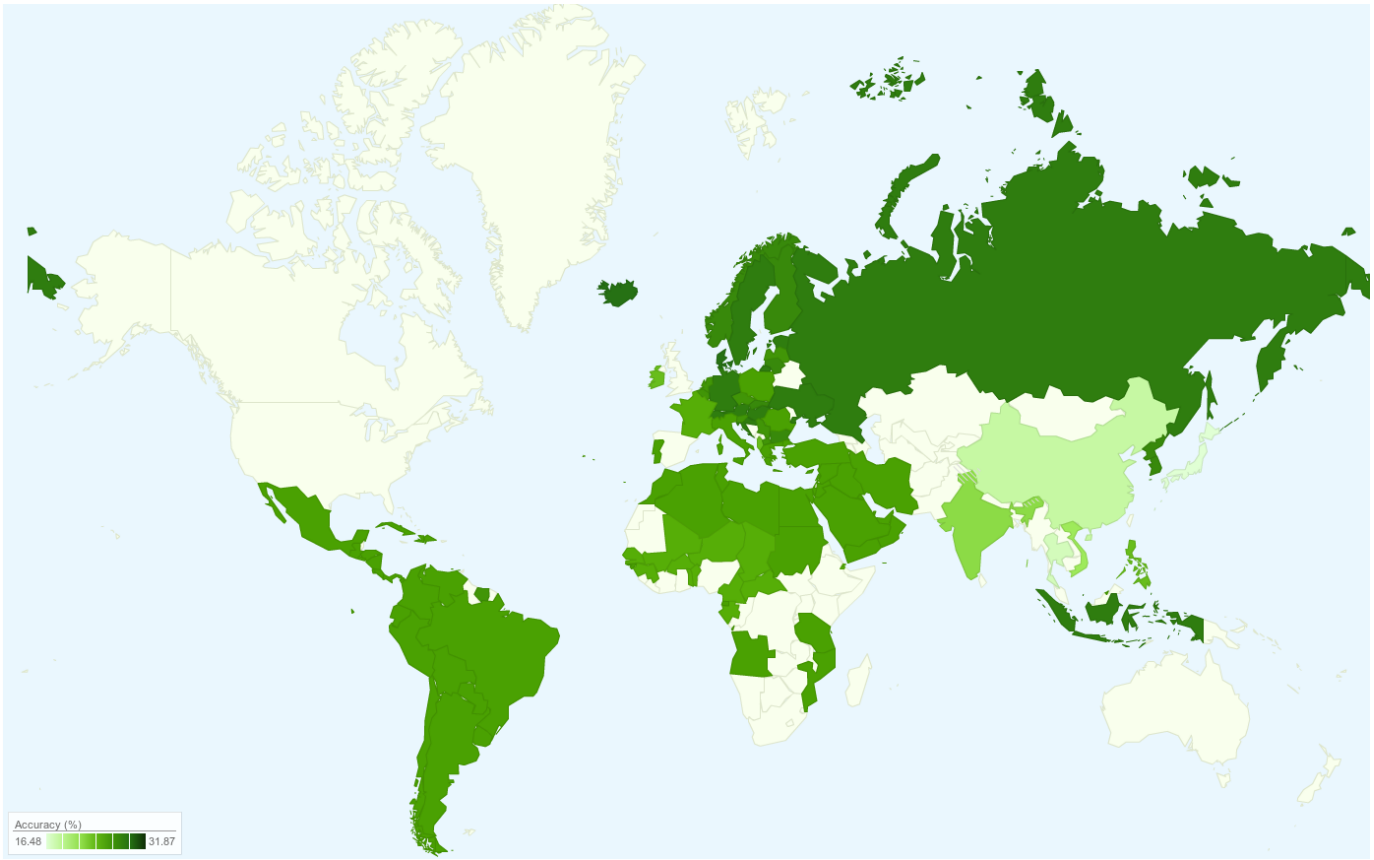
We can detect redundant word with a 37% accuracy

model	accuracy
Random	5.49%
LM	15.38%
round-trip (aligned word)	6.59%
round-trip (exact word match)	6.59%
sig-score	5.49%
Contrib	5.49%
LM+Contrib	**26.37%**
Binary	18.68%
Binary+LM	25.27%
Binary+LM+round-trip (aligned word)	25.27%
Binary+LM+round-trip (exact word match)	23.08%
Binary+LM+sig-score	17.58%
Binary+LM+Contrib	**37.36%**

Our proposed feature **LM+Contrib** works the best in isolation.

Pivot Languages

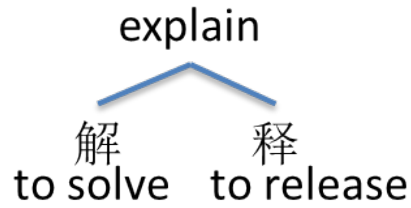
We tried different pivot languages in **LM+Contrib**



European languages generally perform better

Influence from Meaning Components

- Google translate organizes output into characters for Asian languages



- Characters are not the minimum meaning component
- We merged characters/alignments using tokenization result for zh-CN

language	accuracy
de	30.77%
zh-CN	18.68%
zh-CN (char-merged)	29.67%

Fluency v.s. Meaning Redundancy

Two features have different preferences

- **LM** prefers deleting rare content words
- **Contrib** prefers deleting function words

Examples

- illiteracy often limits the$^{\textbf{Contrib}}$ economical$^{\textbf{LM}}$ growth of a nation where knowledge intensive industries are much$^{\textbf{Contrib+LM}}$ highly valued .

The two proposed features add up

model	accuracy
LM	15.38%
Contrib	5.49%
LM+Contrib	**26.37%**

Chapter Summary (Part I)

- We study redundant word detection
  - Redundancy is subjective
  - Determining the most redundant part of a sentence is less subjective
  - We collect a corpus to support our study
- We proposed a machine learning framework for the task
  - We developed features to capture one word's redundancy w.r.t. fluency and meaning
- Our system correctly detect the most redundant word with a 37% accuracy.

Overview of Part II – Correction Detection

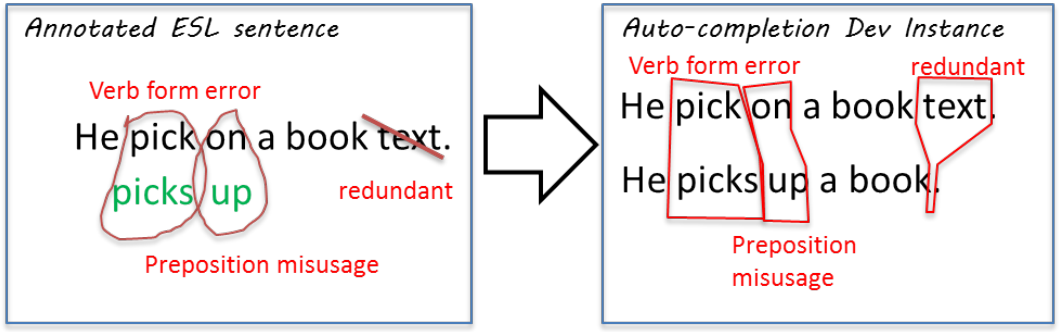
- Introduction
  - Auto-completing language tutors' feedbacks
  - Challenges in correction detection
- Error analysis
- A classifier for merging decisions
- Experiments

Auto-completing Language Tutors' Feedbacks

Auto-completing tutors' feedbacks (Swanson & Yamangil, 2012)

- This can save much efforts, e.g. on Lang-8

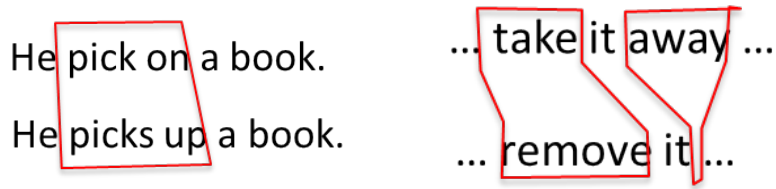
We can develop auto-completion systems on annotated ESL corpora



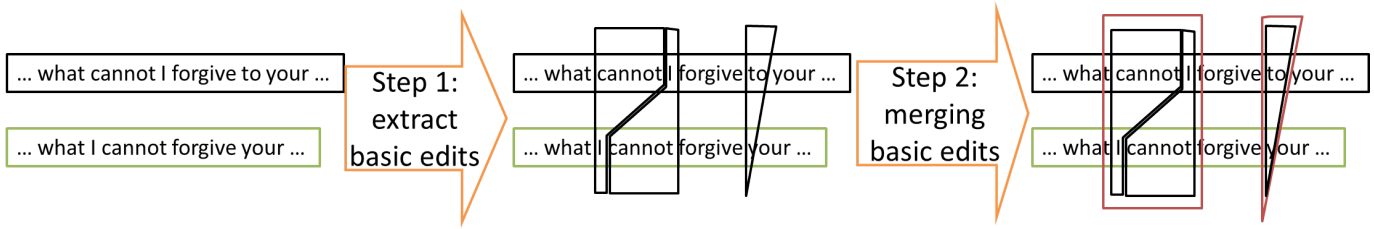
Correction Detection in S&Y

S&Y's system is not perfect: 58% F-score on FCE.

- 70% of system errors occur in correction detection



S&Y's two step correction detection algorithm



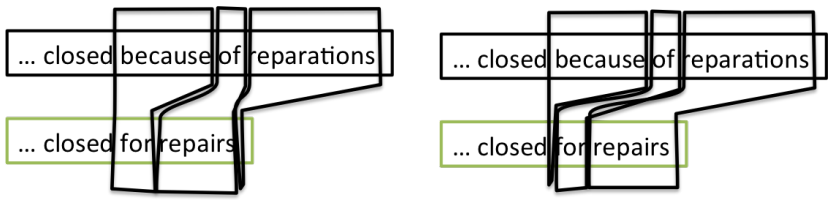
S&Y's heuristics are developed on one single corpus: FCE

- Step 1: Edit-distance
- Step 2: merging adjacent basic edits

Challenge in Correction Detection – Ambiguities

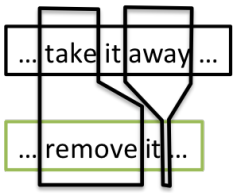
Both steps may incur mistakes.

1. The extracted basic edits might not match our linguistic intuition

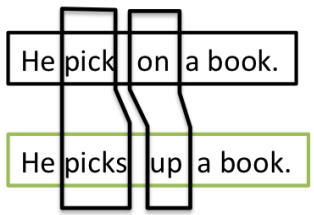


2. Adjacent $\neq$ should merge

non-adjacent but should merge



adjacent but should not merge

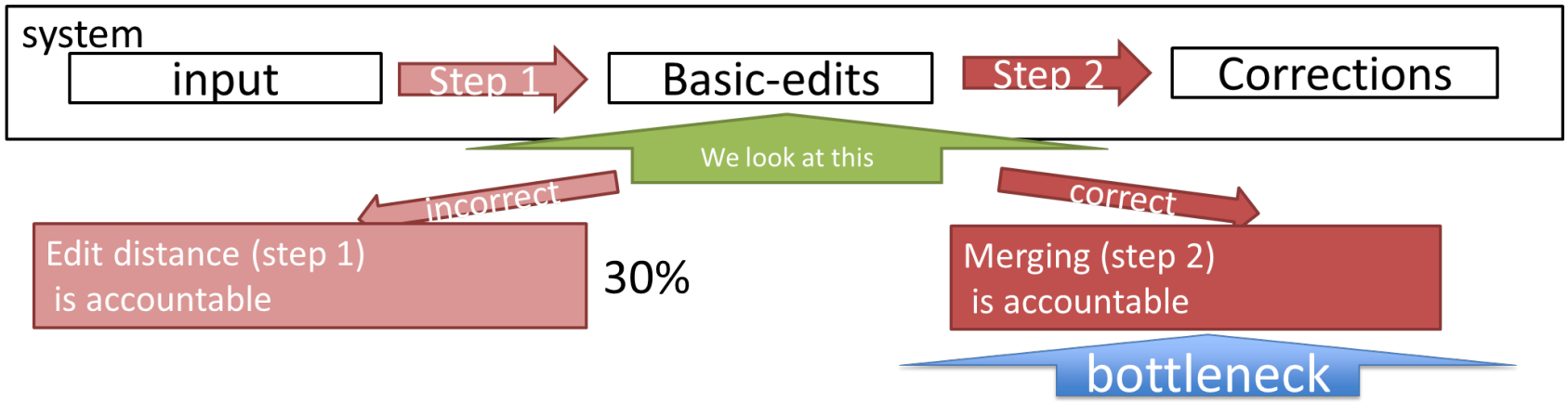


We can develop computational models to reduce these mistakes.

1. Which step is the bottleneck?
2. How can we improve it?

Which Step Causes More Troubles for Computers?

We pick the sentences where S&Y made mistakes. We look at the intermediate step, the set of basic edits.



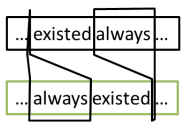
We found the merging step accounts for 70%.

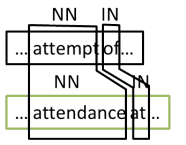
Proposed Classifier for Merging

(Xue and Hwa, ACL 2014)

Intuition: certain patterns indicate whether two edits address the same writing mistake. We encode these patterns into a classifier

- Input: features extracted from two consecutive basic edits
- Output: whether we should merge them



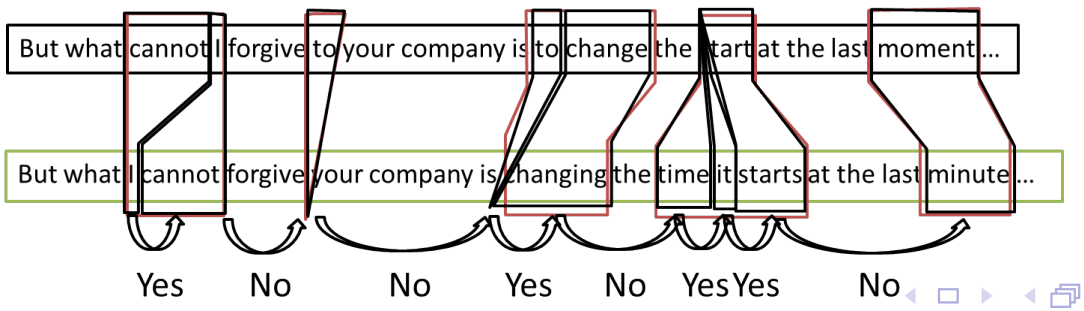


Classifier with Contextual Features	
Does merging the two basic edits match the pattern for a single correction?	gap-between-edits
	Tense-change
	Word-order-error
	Same-word-set
	Revised-to
Does one basic-edit address one single error?	Editdistance=1
	Orig-word-not-in-dict
	Same-POS
	Both-prepositions

Yes

no

Constructing training examples



Questions

1. Can our computational model help auto-complete tutors' feedbacks?
2. Does using additional contextual information help to make better merging decisions?
3. How does our method generalize over different code standards?

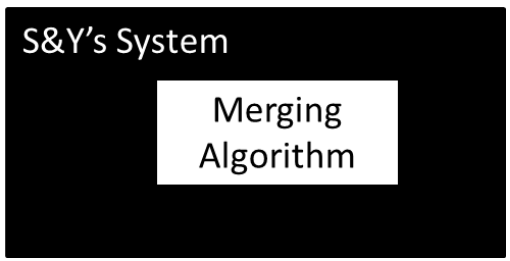
Experimental Setup

We compare different merging algorithms

- S&Y: Merging adjacent basic edits (Swanson & Yamangil, 2012)
- MaxEntMerger: my proposed algorithm

Tasks

- Overall system: Auto-completing the location/reason of individual corrections (Swanson & Yamangil, 2012)



- Intrinsically: correction detection
- Evaluation metric: F_1-score

Corpora: FCE, NUCLE, UIUC, HOO2011

Making Smarter Merging Decisions

Additional contextual information help

Method	Corpus	Correction Detection F_1	Overall F_1-score
S&Y	FCE	70.40%	57.10%
MaxEntMerger	FCE	**80.96%**	**66.36%**
S&Y	NUCLE	61.18%	39.32%
MaxEntMerger	NUCLE	**63.88%**	**41.00%**
S&Y	UIUC	76.57%	65.08%
MaxEntMerger	UIUC	**82.81%**	**70.55%**
S&Y	HOO2011	68.73%	50.95%
MaxEntMerger	HOO2011	**75.71%**	**56.14%**

Our merger led to 10% accuracy improvement for the overall system.

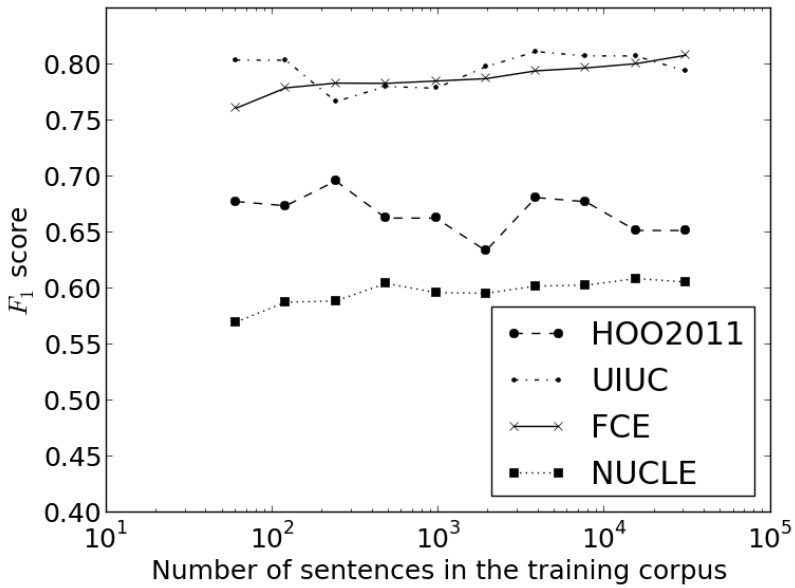
Correction Detection in Different Corpora

The system generalizes over different code standards

training \ testing	FCE	NUCLE	UIUC	HOO2011
S&Y	70.44	61.18%	76.57%	68.73%
FCE	**80.96%**	61.26%	**83.07%**	75.43%
NUCLE	74.53%	**63.88%**	78.57%	74.73%
UIUC	77.25%	58.21%	82.81%	70.83%
HOO2011	71.94%	54.99%	71.19%	**75.71%**

FCE is a comparably good resource for training

- Big data size benefits training



Chapter Summary (Part II)

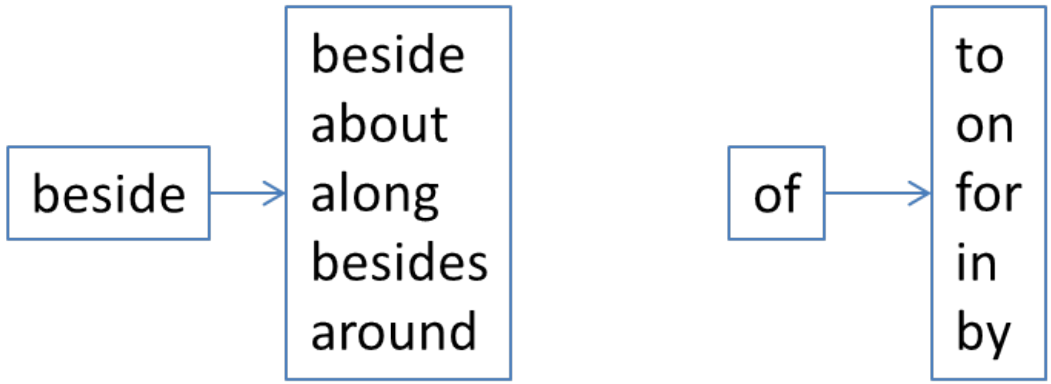
The merging step accounts for 70% errors in correction detection. We propose a merging model:

- Reduces 1/3 errors in correction detection
- Leads to significant overall system performance improvement
- Generalizes over different code standards

Overview of Part III – Confusion Set Construction

- Introduction
  - Confusion Sets
  - Automatic Confusion Set Construction
- Simulation Model
- Experiments

Confusion Sets



Confusion sets are important components for single word mis-usage detection

- They help to rule out unlikely options.
- Limiting the confusion set size help improve GEC system performance (Rozovskaya & Roth, 2010b)
- They were constructed manually (Dahlmeier & Ng, 2011a; Rozovskaya & Roth, 2010b; Liu *et al.*, 2010).

Automatic Confusion Set Construction via Simulation

We propose to construct confusion sets automatically

- Confusions are formed when studying normal English text
- We can
  1. Simulate how learners learn English words
  2. Find out which words are similar

We test our idea on prepositions

- about, along, among, around, as, at, beside, besides, between, by, down, during, except, for, from, in, inside, into, of, off, on, onto, outside, over, through, to, toward, towards, under, underneath, until, up, upon, with, within, without.

Model to Capture Interaction between Meaning and Usage

Learner have understandings about

- Words' out-of-context meanings

for:	on someone's behalf
to:	indicates the purpose

- How a context suggests the meaning

This gift is __ you $\Rightarrow$ The word in between indicates a purpose

While reading: learners adjust their understandings when mismatch

- reading text: . . . this gift is *for* you . . .
  for mis-matches with their expectation *to*
- Words' out-of-context meanings are updated

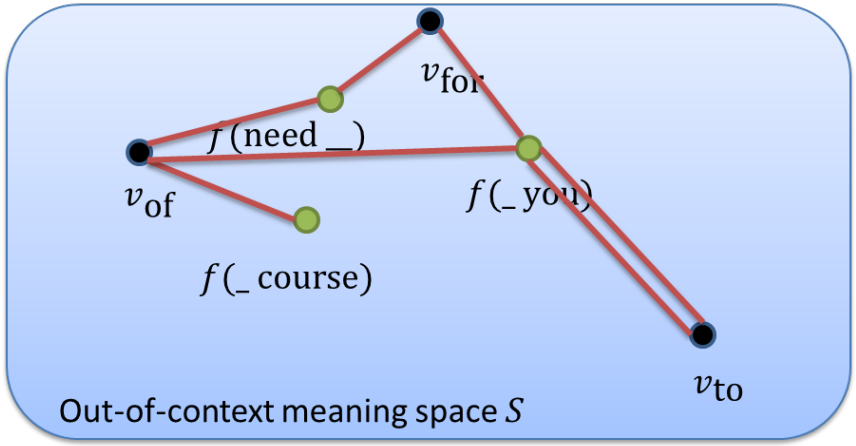
for:	on someone's behalf *or indicates the purpose*
to:	indicates the purpose *in certain contexts*

Confusion Set Construction as an Optimization Problem

(Xue and Hwa, COLING 2012)

I propose a model that infer prepositions' similarities from how they are used.

Corpus
... need of ...
... need for ...
... of you ...
... to you ...
... to you ...
... for you ...
... of course ...
...



Learning goal:

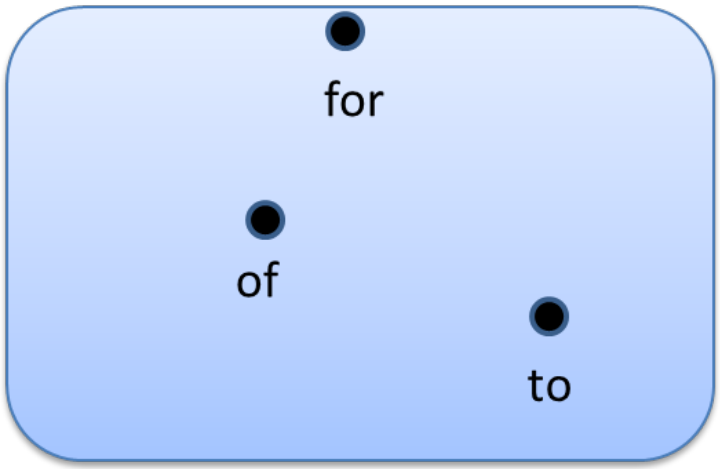
$$\min_{f,\vec{v}_{\text{of}},\ldots} ||f(\text{need\_}) - \vec{v}_{\text{of}}||^2 + \ldots s.t. \text{Area}(\vec{v}_{\text{of}}, \vec{v}_{\text{for}}, \ldots) \geq 1$$

Relevance Component Analysis (RCA) (Bar-Hillel *et al.*, 2006) solves this optimization problem

Using RCA to Construct Confusion Sets

Corpus
... need of ...
... need for ...
... of you ...
... to you ...
... to you ...
... for you ...
... of course ...
...

RCA



Calculate distance

	of	to	for
of	0	2.0	1.5
to	2.0	0	2.5
for	1.5	2.5	0

Pick the closest

Word	Confusion set
of	of, for
to	to, of
for	for, of

Experiments

Question: can we build confusion sets automatically, with the simulation models?
Task: Building Confusion Sets for the 36 most frequent prepositions

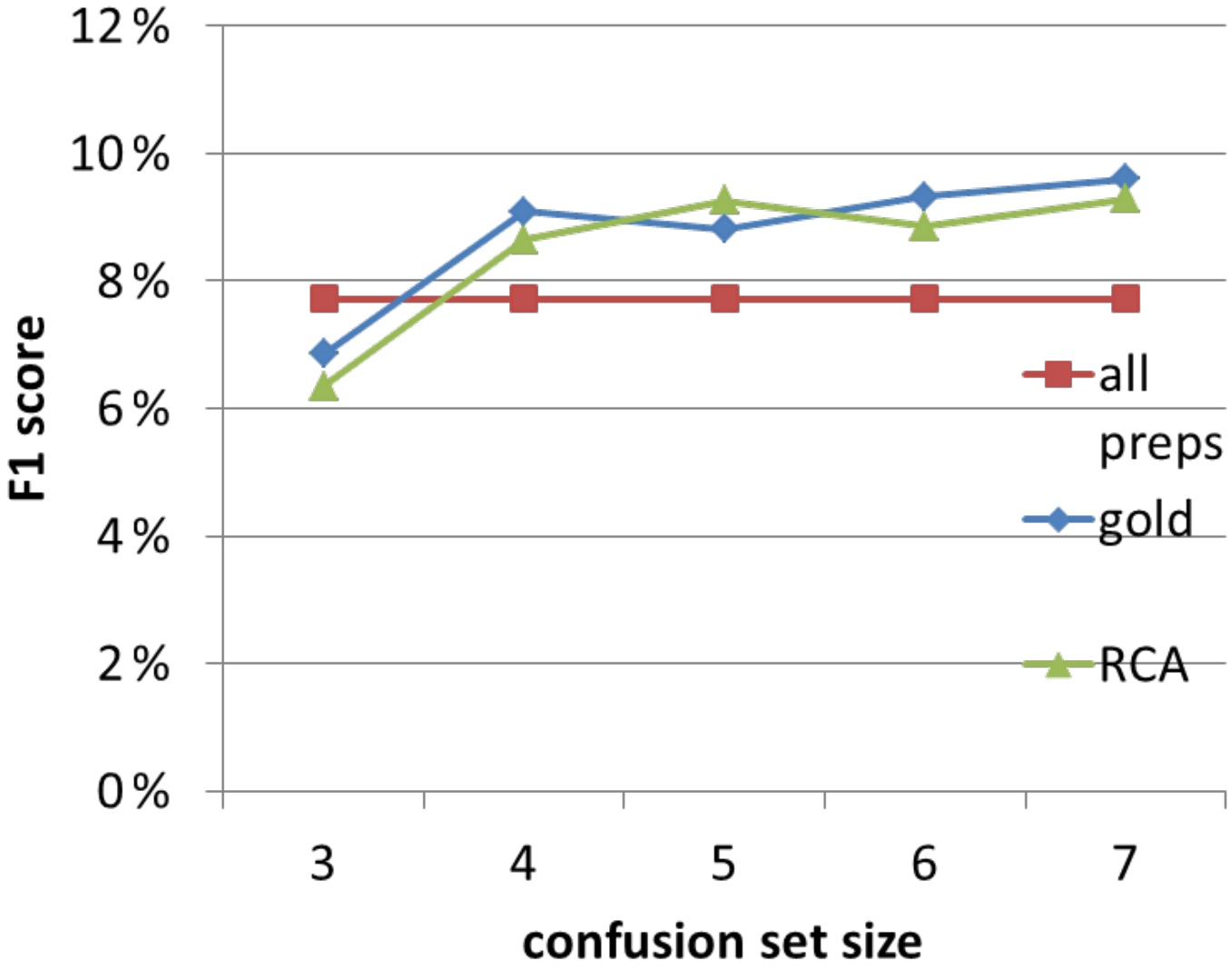
- **RCA**: proposed method, on FBIS (a normal English corpus)
- **all preps**: containing all prepositions
- **gold**: containing the most frequently confused words according to real ESL corpus (NUCLE)

Evaluation: using it in an end-to-end GEC system

- On NUCLE corpus
- Filter training examples using the confusion set (Rozovskaya & Roth, 2010b)
- Evaluate by F_1 score

Experimental Result

Competitive with human annotations in GEC systems



Chapter Summary (Part III)

- Confusion sets are important GEC components
- We propose to build confusion sets via simulating word learning
  - We use RCA algorithm to simulate the interaction between in-context usages and out-of-context meanings
  - We experimented on the 36 most frequent prepositions
  - The automatically constructed confusion sets correlate well with real ESL learners' confusions

Conclusions

- My studies concerned developing computational models to
  1. Highlight error prone areas – redundancy detection
  2. Auto-complete tutors' annotations – correction detection
  3. Eliminate tutors' efforts in building automated systems – automatically building confusion sets
- Computational models can make smart decisions on tutors' behalves.
- This saves tutors' efforts in the long run.

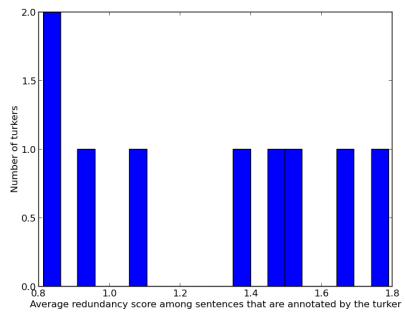
Thank you!

Annotation Study Result

The result suggests that we should

1. automate word level redundancy detection
   - Similar regions are considered more redundant
     - The marked regions overlap on 42% of sentences
     - Among these 42%, one contains the other 83% of the time.
   - Detection result can be helpful for many people: 2/3 sentences were suggested for shortening by ≥ 1 turkers.
2. leave to tutors to decide if the sentence needs shortening
   - It is hard to predict: sentence level kappa is 0.03
   - People have different sensitivities on sentence level

Average sentence-level redundancy score by each turker



References I

Akkaya, Cem, Conrad, Alexander, Wiebe, Janyce, & Mihalcea, Rada. 2010.
Amazon mechanical turk for subjectivity word sense disambiguation.
Pages 195–203 of: Proceedings of the naacl hlt 2010 workshop on creating speech and language data with amazon's mechanical turk.
Association for Computational Linguistics.

Attali, Yigal, & Burstein, Jill. 2006.
Automated essay scoring with e-rater® v. 2.
The journal of technology, learning and assessment, **4**(3).

Bar-Hillel, A., Hertz, T., Shental, N., & Weinshall, D. 2006.
Learning a mahalanobis metric from equivalence constraints.
Journal of machine learning research, **6**(1), 937.

Brown, P.F., Pietra, V.J.D., Pietra, S.A.D., & Mercer, R.L. 1993.
The mathematics of statistical machine translation: Parameter estimation.
Computational linguistics, **19**(2), 263–311.

Clarke, James, & Lapata, Mirella. 2007.
Modelling compression with discourse constraints.
Pages 1–11 of: Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (emnlp-conll).

Cohn, Trevor, Callison-Burch, Chris, & Lapata, Mirella. 2008.
Constructing corpora for the development and evaluation of paraphrase systems.
Computational linguistics, **34**(4), 597–614.

References II

Dahlmeier, D., & Ng, H.T. 2011a.
Correcting semantic collocation errors with l1-induced paraphrases.
Pages 107–117 of: Proceedings of the 2011 conference on empirical methods in natural language processing.
Edinburgh, Scotland, UK: Association for Computational Linguistics.

Dahlmeier, Daniel, & Ng, Hwee Tou. 2011b.
Grammatical error correction with alternating structure optimization.
Pages 915–923 of: Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies - volume 1.
HLT '11.
Portland, Oregon, USA: Association for Computational Linguistics.

Dahlmeier, Daniel, Ng, Hwee Tou, & Wu, Siew Mei. 2013.
Building a large annotated corpus of learner english: The NUS corpus of learner english.
Pages 22–31 of: Proceedings of the eighth workshop on innovative use of nlp for building educational applications.

Dale, Robert, & Kilgarriff, Adam. 2010.
Helping our own: Text massaging for computational linguistics as a new shared task.
Pages 263–267 of: Proceedings of the 6th international natural language generation conference.
Association for Computational Linguistics.

Han, N.R., Tetreault, J., Lee, S.H., & Ha, J.Y. 2010.
Using an error-annotated learner corpus to develop an ESL/efl error correction system.
Lrec, malta, may.

References III

Heilman, Michael, & Smith, Noah A. 2010.
Tree edit models for recognizing textual entailments, paraphrases, and answers to questions.
Pages 1011–1019 of: Human language technologies: The 2010 annual conference of the north american chapter of the association for computational linguistics.
Association for Computational Linguistics.

Hermet, Matthieu, & Désilets, Alain. 2009.
Using first and second language models to correct preposition errors in second language authoring.
Pages 64–72 of: Proceedings of the fourth workshop on innovative use of nlp for building educational applications.
Association for Computational Linguistics.

Koehn, P., Och, F.J., & Marcu, D. 2003.
Statistical phrase-based translation.
Pages 48–54 of: Proceedings of the 2003 conference of the north american chapter of the association for computational linguistics on human language technology-volume 1.
Association for Computational Linguistics.

Leacock, C., Chodorow, M., Gamon, M., & Tetreault, J. 2010.
Automated grammatical error detection for language learners.
Synthesis lectures on human language technologies, **3**(1), 1–134.

Liu, Xiaohua, Han, Bo, Li, Kuan, Stiller, Stephan Hyeonjun, & Zhou, Ming. 2010.
SRL-based verb selection for ESL.
Pages 1068–1076 of: Proceedings of the 2010 conference on empirical methods in natural language processing.
EMNLP '10.
Cambridge, Massachusetts: Association for Computational Linguistics.

References IV

Madnani, Nitin, Tetreault, Joel, & Chodorow, Martin. 2012.
Re-examining machine translation metrics for paraphrase identification.
Pages 182–190 of: Proceedings of the 2012 conference of the north american chapter of the association for computational linguistics: Human language technologies.
Montréal, Canada: Association for Computational Linguistics.

Rozovskaya, Alla, & Roth, Dan. 2010a.
Annotating ESL errors: Challenges and rewards.
Pages 28–36 of: Proceedings of the naacl hlt 2010 fifth workshop on innovative use of nlp for building educational applications.
Association for Computational Linguistics.

Rozovskaya, Alla, & Roth, Dan. 2010b.
Generating confusion sets for context-sensitive error correction.
Pages 961–970 of: Proceedings of the 2010 conference on empirical methods in natural language processing.
EMNLP '10.
Cambridge, Massachusetts: Association for Computational Linguistics.

Snover, Matthew G, Madnani, Nitin, Dorr, Bonnie, & Schwartz, Richard. 2009.
TER-Plus: paraphrase, semantic, and alignment enhancements to translation edit rate.
Machine translation, **23**(2-3), 117–127.

Swanson, Ben, & Yamangil, Elif. 2012.
Correction detection and error type selection as an ESL educational aid.
Pages 357–361 of: Proceedings of the 2012 conference of the north american chapter of the association for computational linguistics: Human language technologies.
Montréal, Canada: Association for Computational Linguistics.

References V

Tetreault, Joel, Foster, Jennifer, & Chodorow, Martin. 2010.
Using parse features for preposition selection and error detection.
Pages 353–358 of: Proceedings of the acl 2010 conference short papers.
ACLShort '10.
Uppsala, Sweden: Association for Computational Linguistics.

Yannakoudakis, Helen, Briscoe, Ted, & Medlock, Ben. 2011.
A new dataset and method for automatically grading esol texts.
Pages 180–189 of: Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1.
Association for Computational Linguistics.