Redundancy Detection in ESL Writings

Huichao Xue and **Rebecca Hwa**
Department of Computer Science,
University of Pittsburgh,
210 S Bouquet St, Pittsburgh, PA 15260, USA
{hux10,hwa}@cs.pitt.edu

Abstract

This paper investigates *redundancy detection* in ESL writings. We propose a measure that assigns high scores to words and phrases that are likely to be redundant within a given sentence. The measure is composed of two components: one captures *fluency* with a language model; the other captures *meaning preservation* based on analyzing alignments between words and their translations. Experiments show that the proposed measure is five times more accurate than the random baseline.

1 Introduction

Writing concisely is challenging. It is especially the case when writing in a foreign language that one is still learning. As a non-native speaker, it is more difficult to judge whether a word or a phrase is redundant. This study focuses on automatically detecting redundancies in English as a Second Language learners' writings.

Redundancies occur when the writer includes some extraneous word or phrase that do not add to the meaning of the sentence but possibly make the sentence more awkward to read. Upon removal of the unnecessary words or phrases, the sentence should improve in its fluency while maintaining the original meaning. In the NUCLE corpus (Dahlmeier and Ng, 2011), an annotated learner corpus comprised of essays written by primarily Singaporean students, 13.71% errors are tagged as "local redundancy errors", making redundancy error the second most frequent problem.[1]

Although redundancies occur frequently, it has not been studied as widely as other ESL errors. A major challenge is that, unlike mistakes that violate the grammaticality of a sentence, redundancies do not necessarily "break" the sentence. Determining which word or phrase is redundant is more of a stylistic question; it is more subjective, and sometimes difficult even for a native speaker.

To the best of our knowledge, this paper reports a first study on redundancy detection. In particular, we focus on the task of defining a redundancy measure that estimates the likelihood that a given word or phrase within a sentence might be extraneous. We propose a measure that takes into account each word's contribution to fluency and meaning. The fluency component computes the language model score of the sentence after the deletion of a word or a phrase. The meaning preservation component makes use of the sentence's translation into another language as pivot, then it applies a statistical machine translation (SMT) alignment model to infer the contribution of each word/phrase to the meaning of the sentence. As a first experiment, we evaluate our measures on their abilities in picking the most redundant phrase of a given length. We show that our measure is five times more accurate than a random baseline.

2 Redundancies in ESL Writings

According to *The Elements of Style* (Strunk, 1918): concise writing requires that *"every word tell."* In that sense, words that "do not tell" are redundant. Determining whether a certain word/phrase is redundant is a stylistic question, which is difficult to quantify. As a result, most annotation resources do not explicitly identify redundancies. One exception is the NUCLE corpus. Below are some examples from the NUCLE corpus, where the bold-faced words/phrases are marked as redundant.

> $\mathbf{Ex_1}$: First of all , there should be a careful consideration about what **are the things that** governments should pay for.

[1] The most frequent error type is *Wrong collocation/idiom preposition*, which comprises 15.69% of the total errors.

Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, pages 683–691,
Gothenburg, Sweden, April 26-30 2014.

$\mathbf{Ex_2}$: GM wishes to reposition itself as an innovative company **to the public**.
$\mathbf{Ex_3}$: These findings are often unpredictable **and uncertain**.
$\mathbf{Ex_4}$: ...the cost incurred is not **only** just large sum of money ...

These words/phrases are considered redundant because they are unnecessary (e.g. $\mathbf{Ex_1}$, $\mathbf{Ex_2}$) or repetitive (e.g. $\mathbf{Ex_3}$, $\mathbf{Ex_4}$).

However, in NUCLE's annotation scheme, some words that were marked redundant are really words that carry undesirable meanings. For example:

$\mathbf{Ex_5}$: ...through which they **can** insert a special ...
$\mathbf{Ex_6}$: ...the analysis and **therefore** selection of a single solution for adaptation...

Note that unlike redundancies, these undesirable words/phrases change the sentences' meanings. Despite the difference in definitions, our experimental work uses the NUCLE corpus because it provides many real world examples of redundancy.

While redundancy detection has not yet been widely studied, it is related to several areas of active research, such as grammatical error correction (GEC), sentence simplification and sentence compression.

Work in GEC attempts to build automatic systems to detect/correct grammatical errors (Leacock et al., 2010; Liu et al., 2010; Tetreault et al., 2010; Dahlmeier and Ng, 2011; Rozovskaya and Roth, 2010). Both redundancy detection and GEC aim to improve students' writings. However, because redundancies do not necessarily break grammaticality, they have received little attention in GEC.

Sentence compression and sentence simplification also consider deleting words from input sentences. However, these tasks have different goals.

Automated sentence simplification (Coster and Kauchak, 2011) systems aim at reducing the grammatical complexity of an input sentence. To illustrate the difference, consider the phrase "critical reception." A sentence simplification system might rewrite it into "reviews"; but a system that removes redundancy should leave it unchanged because neither "critical" nor "reception" is extraneous. Moreover, consider the redundant phrase "had once before" in $\mathbf{Ex_4}$. A simplification system does not need to change it because these words do not add complexity to the sentence.

... and the cost incurred is not only just large sum of money ...

Figure 1: Among the three circled words, "just" is more redundant because deleting it hurts nether fluency nor meaning.

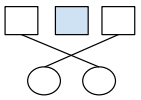
Sentence compression systems (Jing, 2000; Knight and Marcu, 2000; McDonald, 2006; Clarke and Lapata, 2007) aim at shortening a sentence while retaining the most important information and keeping it grammatically correct. This goal distinguishes these systems from ours in two major aspects. First, sentence compression systems assume that the original sentence is well-written; therefore retaining words specific to the sentence (e.g. "uncertain" in $\mathbf{Ex_3}$) can be a good strategy (Clarke and Lapata, 2007). In the ESL context, however, even specific words could still be redundant. For example, although "uncertain" is specific to $\mathbf{Ex_3}$, it is redundant, because its meaning is already implied by "unpredictable". Second, sentence compression systems try to shorten a sentence as much as possible, but an ESL redundancy detector should leave as much of the input sentences unchanged, if possible.

One challenge involved in redundancy detection is that it often involves open class words ($\mathbf{Ex_3}$), as well as multi-word expressions ($\mathbf{Ex_1}$, $\mathbf{Ex_4}$). Current GEC systems dealing with such error types are mostly MT based. MT systems tend to either require large training corpora (Brockett et al., 2006; Liu et al., 2010), or provide whole sentence rewritings (Madnani et al., 2012). Hermet and Désilets (2009) attempted to extract single preposition corrections from whole sentence rewritings. Our work incorporates alignments information to handle complex changes on both word and phrase levels.

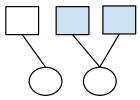
In our approximation, we consider MT output as an approximation of word/phrase meanings. Using words in other languages to represent meanings has been explored in Carpuat and Wu (2007), where the focus is the aligned words' identities. Our work instead focuses more on how many words each word is aligned to.

3 A Probabilistic Model of Redundancy

We consider a word or a phrase to be redundant if deleting it results in a fluent English sentence that conveys the same meaning as before. For example, "not" and "the" are not considered re-



(a) Unaligned English words are considered redundant.



(b) Multiple English words aligned to the same meaning unit. These words are considered redundant.

Figure 2: Configurations our system consider as redundant. In each figure, the shaded squares are the words considered to be more redundant than other words in the same figure.

dundant in Figure 1. This is because discarding "not" would flip the sentence's meaning; discarding "the" would lose a necessary determiner before a noun. In contrast, discarding "just" would hurt neither fluency nor meaning. It is thus considered to be more redundant.

Therefore, our computational model needs to consider words' contributions to both fluency and meaning. Figure 2 illustrates words' contribution to meaning. In those two examples, each subgraph visualizes a sentence: English words correspond to squares in the top row, while their meanings correspond to circles in the bottom row. The knowledge of which word represents what meaning helps in evaluating its contribution. In particular, if a word does not connote any significant meaning, deleting it would not affect the overall sentence; if several words express the same meaning, then deleting some of them might not affect the overall sentence either. Also, deleting a more semantically meaningful word (or phrase) is more likely to cause a loss of meaning of the overall sentence (e.g. *uncertain* v.s. *the*).

Our model computes a single probabilistic value for both fluency judgment and meaning preservation – the log-likelihood that after deleting a certain word or phrase of a sentence, the new sentence is still fluent and conveys the same meaning as before. This value reflects our definition of redundancy – the higher this probability, the more redundant the given word/phrase is.

More formally, suppose an English sentence e contains l_e words: $e = e_1 e_2 \dots e_{l_e}$; after some sub-string $e^{s,t} = e_s \dots e_t (1 \le s \le t \le l_e)$ is deleted from e, we obtain a shorter sentence, denoted as $e_{-}^{s,t}$. We wish to compute the quantity $R(s,t;e)$, the chance that the sub-string $e^{s,t}$ is redundant in sentence e. We propose a probabilistic model to formalize this notion.

Let M be a random variable over some meaning representation; $\Pr(M|e)$ is the likelihood that M carries the meaning of e. If the sub-string $e^{s,t}$ is redundant, then the new sentence $e_{-}^{s,t}$ should still express the same meaning; $\Pr(e_{-}^{s,t}|M)$ computes the likelihood that the after-deletion sentence can be generated from meaning M.

$$
\begin{aligned}
& R(s,t;e) \\
= \; & \log \sum_{M=m} \Pr(m|e) \Pr(e_{-}^{s,t}|m) \\
= \; & \log \sum_{M=m} \frac{\Pr(m|e) \Pr(e_{-}^{s,t}) \Pr(m|e_{-}^{s,t})}{\Pr(m)} \\
= \; & \log \Pr(e_{-}^{s,t}) + \log \sum_{M=m} \frac{\Pr(m|e_{-}^{s,t}) \Pr(m|e)}{\Pr(m)} \\
= \; & \mathrm{LM}(e_{-}^{s,t}) + \mathrm{AGR}(M|e_{-}^{s,t}, e) \qquad (1)
\end{aligned}
$$

The first term $\mathrm{LM}(e_{-}^{s,t})$ is the after-deletion sentence's log-likelihood, which reflects its fluency. We calculate the first term with a trigram language model (LM).

The second term $\mathrm{AGR}(M|e_{-}^{s,t}, e)$ can be interpreted as the chance that e and $e_{-}^{s,t}$ carry the same meaning, discounted by "chance agreement". This term captures meaning preservation.

The two terms above are complementary to each other. Intuitively, LM prefers keeping common words in $e_{-}^{s,t}$ (e.g. *the, to*) while AGR prefers keeping words specific to e (e.g. *disease, hypertension*).

To make the calculation of the second term practical, we make two simplifying assumptions.

Assumption 1 A sentence's meaning can be represented by its translations in another language; its words' contributions to the meaning of the sentence can be represented by the mapping between the words in the original sentence and its translations (Figure 3).

Note that the choice of translation language may impact the interpretation of words' contributions. We will discuss about this issue in our experiments (Section 5).

Assumption 2 Instead of considering all possible translations f for e, our computation will make use of the most likely translation, f_*.

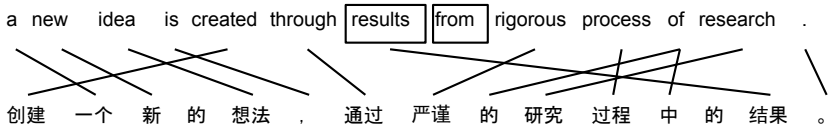


Figure 3: Illustration of Assumption 1 and Approximation 1. An English sentence's meaning is presented as a Chinese translation. Meanwhile, each (English) word's contribution to the sentence meaning is realized as a word alignment. For Approximation 1, note that sentence alignments normally won't be affected before/after deleting words (e.g. "results from") from the source sentence.

With the two approximations:

$$\begin{aligned}\mathrm{AGR}(M|e_{-}^{s,t},e) &\approx \log\frac{\Pr(f_*|e_{-}^{s,t})\Pr(f_*|e)}{\Pr(f_*)}\\ &= \log\Pr(f_*|e_{-}^{s,t})+C_1(e)\end{aligned}$$

(We use $C_i(e)$ to denote constant numbers within sentence e throughout the paper.)

We now rely on a statistical machine translation model to approximate the translation probability $\log\Pr(f_*|e_{-}^{s,t})$.

One naive way of calculating this probability measure is to consult the MT system. This method, however, is too computationally expensive for one single input sentence. For a sentence of length n, calculating the redundancy measure for all chunks in it would require issuing $O(n^2)$ translation queries. We propose an approximation that instead calculates the difference of translation probability caused by discarding $e^{s,t}$, based on an analysis on the alignment structure between e and f_*. We show the measure boils down to summing the expected number of aligned words for each $e_i(s\le i\le t)$, and possibly weighting these numbers by e_i's unigram probability. This method requires one translation query, and $O(n^2)$ queries into a language model, which is much more suitable for practical applications. Our method also sheds light on the role of alignment structures in the redundancy detection context.

3.1 Alignments Approximation

One key insight in our approximation is that the alignment structure a between $e_{-}^{s,t}$ and f_* would be largely similar with the alignment structure between e and f_*. We illustrate this notion in Figure 3. Note that after deleting two words "results from" from the source sentence in Figure 3, the alignment structure remains unchanged elsewhere. Also, "结果", the word once connected with "results", can now be seen as connected to blanks.

We hence approximate $\log\Pr(f_*|e_{-}^{s,t})$ by reusing the alignment structure between e and f_*. To make the alignment structures compatible, we start with redefining $e_{-}^{s,t}$ as $e_1,e_2,\ldots,e_{s-1},\square,\ldots,\square,e_{t+1},\ldots,e_{l_e}$, where the deleted words are left blank.

Let $\Pr(a|f,e)$ be the posterior distribution of alignment structure between sentence pair (f,e).

Approximation 1 We formalize the similarity between the alignment structures by assuming the KL-divergence between their alignment distributions to be small.

$$D_{\mathrm{KL}}(a|f_*,e;a|f_*,e_{-}^{s,t})\approx 0$$

This allows using $\Pr(a|f_*,e)$ to help approximate $\log\Pr(f_*|e_{-}^{s,t})$:

$$\begin{aligned}&\log\Pr(f_*|e_{-}^{s,t})\\ =\;&\log\sum_a\Pr(a|f_*,e)\frac{\Pr(f_*,a|e_{-}^{s,t})}{\Pr(a|f_*,e)}\\ =\;&\sum_a\Pr(a|f_*,e)\log\frac{\Pr(f_*,a|e_{-}^{s,t})}{\Pr(a|f_*,e)}\\ &+\underbrace{\sum_a\Pr(a|f_*,e)\log\left(\Pr(f_*|e_{-}^{s,t})/\frac{\Pr(f_*,a|e_{-}^{s,t})}{\Pr(a|f_*,e)}\right)}_{D_{\mathrm{KL}}(a|f_*,e;a|f_*,e_{-}^{s,t})\approx 0}\\ \approx\;&\sum_a\Pr(a|f_*,e)\log\Pr(f_*|e_{-}^{s,t},a)+C_2(e)\end{aligned}$$

We then use an SMT model to calculate $\log\Pr(f_*|e_{-}^{s,t},a)$, the translation probability under a given alignment structure.

3.2 The Translation Model

Approximation 2 We will use IBM Model 1 (Brown et al., 1993) to calculate $\log\Pr(f_*|e_{-}^{s,t},a)$

IBM Model 1 is one of the earliest statistical translation models. It helps us to compute

$\log \Pr(f_*|e_-^{s,t}, a)$ by making explicit how each word contributes to words it aligns with. In particular, to compute the probability that f is a translation of e, $\Pr(f|e)$, IBM Model 1 defined a generative alignment model where every word f_i in f is aligned with exactly one word e_{a_i} in e, so that f_i and e_{a_i} are word level translations of each other.

$$\begin{aligned}
&\sum_a \Pr(a|f_*, e) \log \Pr(f_*|e_-^{s,t}, a) \\
= &\sum_a \Pr(a|f_*, e) \sum_{1 \le i \le l_{f_*}} \log \Pr(f_*^i|e_{-a_i}^{s,t}) \\
= &\sum_a \Pr(a|f_*, e) \sum_{1 \le i \le l_{f_*}} \log \frac{\Pr(f_*^i|e_{-a_i}^{s,t})}{\Pr(f_*^i|e_{a_i})} + C_3(e)
\end{aligned}$$

Note that

$$\log \frac{\Pr(f_*^i|e_{-a_i}^{s,t})}{\Pr(f_*^i|e_{a_i})} = \begin{cases} 0 & \text{, for } a_i \notin \{s \dots t\} \\ \log \frac{\Pr(f_*^i|\square)}{\Pr(f_*^i|e_{a_i})} & \text{, otherwise} \end{cases}$$

$$\begin{aligned}
&\sum_a \Pr(a|f_*, e) \sum_{1 \le i \le l_{f_*}} \log \frac{\Pr(f_*^i|e_{-a_i}^{s,t})}{\Pr(f_*^i|e_{a_i})} \\
= &\sum_a \Pr(a|f_*, e) \sum_{1 \le i \le l_{f_*}} \sum_{s \le j \le t} I_{a_i=j} \log \frac{\Pr(f_*^i|\square)}{\Pr(f_*^i|e_j)} \\
= &\underbrace{\sum_{s \le j \le t} \sum_{1 \le i \le l_{f_*}} \underbrace{\Pr(a_i = j|f_*, e)}_{A_{i,j}} \log \frac{\Pr(f_*^i|\square)}{\Pr(f_*^i|e_j)}}_{\mathrm{DIFF}(e_-^{s,t}, e)}
\end{aligned}$$

Here $A_{i,j} = \Pr(a_i = j \mid f_*, e)$, which is the probability of the i-th word in the translation being aligned to the j-th word in the original sentence.

3.3 Per-word Contribution

Through deductions,

$$\begin{aligned}
R(s,t;e) &= \mathrm{LM}(e_-^{s,t}) + \mathrm{DIFF}(e_-^{s,t}, e) \\
&\quad + C_1(e) + C_2(e) + C_3(e)
\end{aligned}$$

the redundancy measure boils down to how we define $\Pr(f_*^i|\square_j)$, which is: when we discard e_j, how do we generate the word it aligns f_*^i with in its translation. This value reflects e_j's contribution in generating f_*^i.

We approximate $\Pr(f_*^i|\square_j)$ in two ways.

1. Suppose that all words in the translation are of equal importance. We assume $\log \frac{\Pr(f_*^i|\square)}{\Pr(f_*^i|e_j)} = -C_c$, where C_c is a constant number. A larger C_c value indicates a higher importance of e_j during the translation.

$$\begin{aligned}
\mathrm{DIFF}(e_-^{s,t}, e) &= -C_c \sum_{s \le j \le t} \sum_{1 \le i \le l_{f_*}} A_{i,j} \\
&= -C_c \sum_{s \le j \le t} A(j) \qquad (2)
\end{aligned}$$

Here $A(j)$ is the expected number of alignments to e_j. This metric demonstrates the intuition that words aligned to more words in the translation are less redundant.

2. We note that rare words are often more important, and therefore harder to be generated. We assume $\Pr(f_*^i|\square) = \Pr(e_j|\square) \Pr(f_*^i|e_j)$.

$$\begin{aligned}
&\mathrm{DIFF}(e_-^{s,t}, e) \\
= &\sum_{s \le j \le t} \sum_{1 \le i \le l_{f_*}} A_{i,j} \log \frac{\Pr(e_j|\square) \Pr(f_*^i|e_j)}{\Pr(f_*^i|e_j)} \\
= &\sum_{s \le j \le t} A(j) \log \Pr(e_j|\square) \qquad (3)
\end{aligned}$$

This gives us counts on how likely each word is aligned with Chinese words according to $\Pr(a|f_*, e)$, where each word is weighted by its importance $\log \Pr(e_j|\square)$. We use e_j's unigram probability to approximate $\log \Pr(e_j|\square)$.

When estimating the alignment probabilities $A_{i,j}$, we smooth the alignment result from Google translation using Dirichlet-smoothing, where we set $\alpha = 0.1$ empirically based on experiments in the development dataset.

4 Experimental Setup

A fully automated redundancy detector has to decide (1) whether a given sentence contains any redundancy errors; (2) how many words constitute the redundant part; and (3) which exact words are redundant. In this paper, we focus on the third part while assuming the first two are given. Thus, our experimental task is: given a sentence known to contain a redundant phrase of a particular length, can that redundant phrase be accurately identified? For most sentences in our study, this results in choosing one from around 20 words/phrases.

While the task has a somewhat limited scope, it allows us to see how we could formally measure the difference between redundant words/phrases

and non-redundant ones. For each measure, we observe whether it has assigned the highest score to the redundant part of the sentence. We compare the proposed redundancy model described in Section 3 against a set of baselines and other potential redundancy measures (to be described shortly).

To better understand different measures' performance on function words vs. content words, we also calculate the percentage of redundant function/content words that are detected successfully – accuracy in both categories. In our experiments, we consider prepositions and determiners as function words; and we consider other words/phrases as content words/phrases.

4.1 Redundancy Measures

To gain insight into redundancy error detection's difficulty, we first consider a random baseline.

random The random baseline assigns a random score to each word/phrase. The resulting system will pick one word/phrase of the given length at random.

We consider relying on large scale language models to decide redundancy.

trigram We use a trigram language model to capture fluency, by calculating the log-likelihood of the whole sentence after discarding the given word/phrase. A higher probability indicates a higher fluency.

round-trip Inspired by Madnani et al. (2012; Hermet and Désilets (2009), an MT system may eliminate grammar errors with the help of large scale language models. In this method, we analyze which parts are considered redundant by an MT system by comparing the original sentence with its round-trip translation. We use Google translate to first translate one sentence into a pivot language, and then back to English. We measure one phrase's redundancy by the number of words that disappeared after the round-trip. We determine if one word disappeared in two ways:

extract word match: one word is considered disappeared if the same word does not occur in the round-trip.

aligned word: we use the Berkeley aligner (DeNero and Klein, 2007) to align original sentences with their round-trip translations. Unaligned words are considered to have disappeared.

We consider measures for words/phrases' contributions to sentence meaning.

sig-score This measure accounts for whether one word w_i is capturing the gist of a sentence (Clarke and Lapata, 2007)[2]. It was shown to help decide whether one part should be discarded during sentence compression.

$$I(w_i) = -\frac{l}{N} \cdot f_i \log \frac{F_a}{F_i}$$

f_i and F_i are the frequencies of w_i in the current document and a large corpus respectively; F_a is the number of all word occurrences in the corpus; l is the number of clause constituents above w_i; N is the deepest level of clause embeddings. This measure assigns low scores to document specific words occurring at deep syntax levels.

align # We use the number of alignments that a word/phrase has in the translation to measure its redundancy, as deducted in Equation 2.

contrib We compute the word/phrase's contribution to meaning, according to Equation 3.

We consider the combinations of measures.

trigram + $C_c$align # We use a linear combination between language model and align # (Equation 2). We tune C_c on development data.

trigram+contrib This measure (as we proposed in Section 3) is the sum of the *trigram* language model component and the *contrib* component which represents the phrase's contribution to meaning.

trigram+α round-trip/sig-score We combine language model with **round-trip** and **sig-score** linearly (McDonald, 2006; Clarke and Lapata, 2007). To obtain baselines that are as strong as possible, we tune the weight α on evaluation data for best accuracy.

4.2 Pivot Languages

Our proposed model uses machine translation outputs from different pivot languages. To see which language helps measuring redundancy, we compare 52 pivot languages available at Google translate[3] for meaning representation[4].

[2]We extend this measure, which was only defined for content words in Clarke and Lapata (2007), to include all English words.

[3]http://translate.google.com

[4]These languages include Albanian (sq), Arabic (ar), Azerbaijani (az), Irish (ga), Estonian (et), Basque (eu),

length	count	percentage
1	356	67.55%
2	80	15.18%
3	40	7.59%
4	18	3.42%
other	33	6.26%

Table 1: Length distribution of redundant chunks' lengths in the evaluation data.

4.3 Data and Tools

We extract instances from the NUCLE corpus (Dahlmeier and Ng, 2011), an error annotated corpus mainly written by Singaporean students, to conduct this study. The corpus is composed of 1,414 student essays on various topics. Annotations in NUCLE include error locations, error types, and suggested corrections. Redundancy errors are marked by annotators as *Rloc*. In this study, we only consider the cases where the suggested correction is to delete the redundant part (97.09% among all *Rloc* errors).

To construct our evaluation dataset, we pick sentences with exactly one redundant word/phrase. This is the most common case (81.18%) among sentences containing redundant words/phrases. We use 10% of the essays (336 sentences) for development purposes, and another 200 essays as the evaluation corpus (527 sentences). A distribution of redundant chunks' lengths in evaluation corpus is shown in Table 1.

We train a trigram language model using the SRILM toolkit (Stolcke, 2002) on the Agence France-Presse (afp) portion of the English Gigawords corpus.

5 Experiments

The experiment aims to address the following questions: (1) Does a sentence's translation serve as a reasonable approximation for its meaning? (2) If so, does the choice of the pivot language matter? (3) How do the potentially conflicting goals of preserving fluency versus preserving meaning impact the definition of a redundancy measure?

Our experimental results are presented in Figure 4 and Table 2. In Figure 4 we compare using different pivot languages in our proposed model; in Table 2 we compare using different redundancy metrics for the same pivot language – French.

Metrics	overall	function words	content words
random	4.44%	4.62%	4.36%
trigram	8.06%	3.95%	9.73%
sig-score	10.71%	22.16%	6.07%
round-trip (aligned word)	10.69%	12.72%	9.87%
round-trip (exact word match)	5.75%	4.27%	6.35%
trigram + α round-trip (aligned word)	14.80%	11.84%	16.00%
trigram + α round-trip (exact word match)	9.49%	4.61%	11.47%
trigram + α sig-score	11.01%	22.68%	6.28%
align #	5.04%	3.36%	5.72%
trigram + $C_c \times$align #	9.58%	4.61%	11.60%
contrib	8.59%	20.23%	3.87%
trigram + contrib	21.63%	38.16%	14.93%

Table 2: Redundancy part identification accuracies for different redundancy metrics on NUCLE corpus, using French as the pivot language.

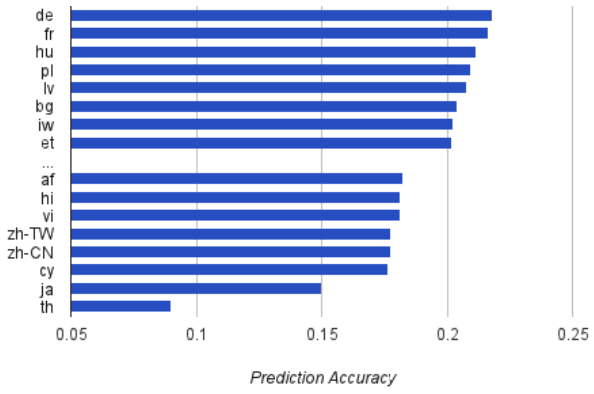


Figure 4: Using different pivot languages for redundancy measurement.

First, compared to other measures, our proposed model best captures redundancy. In particular, our model picks the correct redundant chunk 21.63% of the time, which is five times higher than the random baseline. This suggests that using translation to approximate sentence meanings is a plausible option. Note that one partial reason for the low figures is the limitation of data resources. During error analysis, we found linkers/connectors (e.g. moreover, however) and modal auxiliaries (e.g.

Byelorussian (be), Bulgarian (bg), Icelandic (is), Polish (pl), Persian (fa), Boolean (language ((Afrikaans) (af), Danish (da), German (de), Russian (ru), French (fr), Tagalog (tl), Finnish (fi), Khmer (km), Georgian (ka), Gujarati (gu), Haitian (Creole (ht), Korean (ko), Dutch (nl), Galician (gl), Catalan (ca), Czech (cs), Kannada (kn), Croatian (hr), Latin (la), Latvian (lv), Lao (lo), Lithuanian (lt), Romanian (ro), Maltese (mt), Malay (ms), Macedonian (mk), Bengali (bn), Norwegian (no), Portuguese (pt), Japanese (ja), Swedish (sv), Serbian (sr), Esperanto (eo), Slovak (sk), Slovenian (sl), Swahili (sw), Telugu (te), Tamil (ta), Thai (th), Turkish (tr), Welsh (cy), Urdu (ur), Ukrainian (uk), Hebrew (iw), Greek (el), Spanish (es), Hungarian (hu), Armenian (hy), Italian (it), Yiddish (yi), Hindi (hi), Indonesian (id), English (en), Vietnamese (vi), Simplified Chinese (zh-CN), Traditional Chinese (zh-TW).

can, had) are often marked redundant when they actually carry undesirable meanings ($\mathbf{Ex_6}$, $\mathbf{Ex_5}$). These cases comprise a 16% portion among our model's failures. Despite this limitation, the evaluation still suggests that current approaches are not ready for a full redundancy detection pipeline.

Second, we find that the choice of pivot language does make a difference. Experimental result suggests that the system tends to achieve higher redundancy detection accuracy when using translations of a language more similar to English. In particular, when using European languages (e.g. German (de), French (fr), Hungarian (hu) etc.) as pivot, the system performs much better than using Asian languages (e.g. Chinese (zh-CN), Japanese (ja), Thai (th) etc.). One reason for this phenomenon is that the default Google translation output in Asian languages (as well as the alignment between English and these languages) are organized into characters, while characters are not the minimum meaning component. For example, in Chinese, "解释" is the translation of "explanation", but the two characters "解" and "释" mean "to solve" and "to release" respectively. In the alignment output, this will cause certain words being associated with more or less alignments than others. In this case, the number of alignments no longer directly reflect how many meaning units a certain word helps to express. To confirm this phenomenon, we tried improving the system using Simplified Chinese as the pivot language by merging characters together. In particular, we applied Chinese tokenization (Chang et al., 2008), and then merged alignments accordingly. This raised the system's accuracy from 17.74% to 20.11%.

Third, to better understand the salient features of a successful redundancy measure, we experimented with using different components in isolation. We find that the language model component is better at detecting redundant content words, while the alignment analysis component is better at detecting redundant function words. The language model detects the function word redundancies with a worse accuracy than the random baseline; the alignment analysis component also has a worse accuracy than the random baseline on content words. However, the English language model and the alignment analysis result can build on top of each other when we analyze the redundancies.

We also found that alignments help us to better account for each word's contribution to the "meaning" of the sentence. A linear combination of a language model score and our proposed measure based on analysis of alignments best captures redundancy. However, as our experimental results suggest, it is necessary both to use alignments in translation outputs, and to use them in a good way. Alignments help isolating fluency from the meaning component – making them easy to integrate. As our experiments demonstrated, although methods comparing Google round-trip translation's output with the original sentence could lead to a 10.69% prediction accuracy, it is harder to combine it with the English language model. This is partly because of the non-orthogonality of these two measures – the English language model has already been used in the round-trip translation result. Also, an information theoretical interpretation of alignments is essential for the model's success. For example, a more naive way of using alignment results, **align #,** which counts the number of alignments, leads to a much lower accuracy.

6 Conclusions

Despite the prevalence of redundant phrases in ESL writings, there has not been much work in the automatic detection of these problems. We conduct a first study on developing a computational model of redundancies. We propose to account for words/phrases redundancies by comparing an ESL sentence with outputs from off-the-shelf machine translation systems. We propose a redundancy measure based on this comparison. We show that by interpreting the translation outputs with IBM Models, redundancies can be measured by a linear combination of a language model score and the words' contribution to the sentence's meaning. This measure accounts for both the fluency and completeness of a sentence after removing one chunk. The proposed measure outperforms the direct round-trip translation and a random baseline by a large margin.

Acknowledgements

This work is supported by U.S. National Science Foundation Grant IIS-0745914. We thank the anonymous reviewers for their suggestions; we also thank Joel Tetreault, Janyce Wiebe, Wencan Luo, Fan Zhang, Lingjia Deng, Jiahe Qian, Nitin Madnani and Yafei Wei for helpful discussions.

References

Chris Brockett, William B. Dolan, and Michael Gamon. 2006. Correcting ESL errors using phrasal smt techniques. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, ACL-44, pages 249–256, Sydney, Australia. Association for Computational Linguistics.

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

M. Carpuat and D. Wu. 2007. Improving statistical machine translation using word sense disambiguation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 61–72.

Pi-Chuan Chang, Michel Galley, and Christopher D Manning. 2008. Optimizing chinese word segmentation for machine translation performance. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 224–232. Association for Computational Linguistics.

James Clarke and Mirella Lapata. 2007. Modelling compression with discourse constraints. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 1–11.

Will Coster and David Kauchak. 2011. Learning to simplify sentences using wikipedia. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, pages 1–9, Portland, Oregon, June. Association for Computational Linguistics.

Daniel Dahlmeier and Hwee Tou Ng. 2011. Grammatical error correction with alternating structure optimization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 915–923, Portland, Oregon, USA. Association for Computational Linguistics.

John DeNero and Dan Klein. 2007. Tailoring word alignments to syntactic machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 17–24, Prague, Czech Republic, June. Association for Computational Linguistics.

Matthieu Hermet and Alain Désilets. 2009. Using first and second language models to correct preposition errors in second language authoring. In *Proceedings of the Fourth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 64–72. Association for Computational Linguistics.

Hongyan Jing. 2000. Sentence reduction for automatic text summarization. In *Proceedings of the sixth conference on Applied natural language processing*, pages 310–315. Association for Computational Linguistics.

Kevin Knight and Daniel Marcu. 2000. Statistics-based summarization - step one: Sentence compression. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pages 703–710. AAAI Press.

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

Xiaohua Liu, Bo Han, Kuan Li, Stephan Hyeonjun Stiller, and Ming Zhou. 2010. SRL-based verb selection for ESL. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 1068–1076, Cambridge, Massachusetts. Association for Computational Linguistics.

Nitin Madnani, Joel Tetreault, and Martin Chodorow. 2012. Re-examining machine translation metrics for paraphrase identification. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 182–190, Montréal, Canada, June. Association for Computational Linguistics.

Ryan McDonald. 2006. Discriminative sentence compression with soft syntactic evidence. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, volume 6, pages 297–304. Association for Computational Linguistics.

Alla Rozovskaya and Dan Roth. 2010. Generating confusion sets for context-sensitive error correction. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 961–970, Cambridge, Massachusetts. Association for Computational Linguistics.

Andreas Stolcke. 2002. Srilm-an extensible language modeling toolkit. In *Proceedings of the international conference on spoken language processing*, volume 2, pages 901–904.

William Strunk. 1918. *The elements of style / by William Strunk, Jr.*

Joel Tetreault, Jennifer Foster, and Martin Chodorow. 2010. Using parse features for preposition selection and error detection. In *Proceedings of the ACL 2010 Conference Short Papers*, ACLShort '10, pages 353–358, Uppsala, Sweden. Association for Computational Linguistics.