Idiomify: Building a collocation-supplemented reverse-dictionary of idioms with Word2Vec for L2 learners

- author: Eu-Bin KIM
- the deadline is today. By 12pm. That is a hard deadline.
- expected word count: 13000.

- intro: 0, 0, 0
- methods: 0, 0, 0
- results & discussions: 0, 0, 0
- write as much as you can. That's the goal
- keep it simple.. right?

Abstract

- conclusion: solves tip-of-the-tongue problem, and helps L2 learners use idioms precisely and naturally.

Table of Contents

- Abstract (500)

1. The aim (500)
2. introduction (3200)
3. objective (400)
    - Just communicate **what it does** right off the bat.
    - what example could I use for this?
4. motivation (1250)
    - why idiomify?
    - why idiom2collocations?
5. related work (1250)
    - There are rule-based approaches & ml-based approaches.
6. aims (restated) (400)
7. methods and justifications (4000)
8. identifying idioms - what & why? - how? - how well does it work? -> links to the results

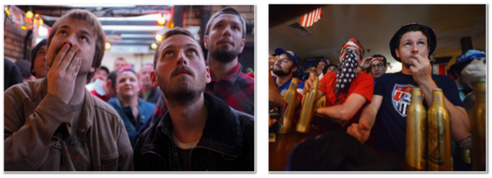
9. preprocessing data - this should be short. - what & why? - how? - how well does it work?

10. modeling & extracting collocation - what & why? (links back to 1.2) - how? - how well does it work?

   - It is difficult to evaluate this quantitatively, so we evaluate this quantitatively.
     - how could you improve upon this?
   - x__pearson,
11. training idiom2vec - what & why? - how? - how well does it work? - how could you improve upon this?
12. idiomifying a phrase - what & why? - how? - how well does it work? - how could you improve upon this?
13. results and discussions (4000)
14. idiom2collocations - good side? - bad side? - critical points.

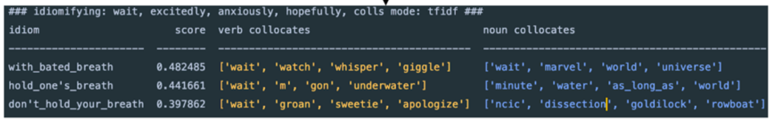
15. idiom2vec
16. idiomify
17. References (500)

1. The aim



They are waiting excitedly, anxiously and hopefully to see the results.

Idiomify



```
### idiomifying: wait, excitedly, anxiously, hopefully, colls mode: tfidf ###
idiom                    score  verb collocates                              noun collocates
---------------------  --------  -----------------------------------------  ----------------------------------------------
with_bated_breath      0.482485  ['wait', 'watch', 'whisper', 'giggle']     ['wait', 'marvel', 'world', 'universe']
hold_one's_breath      0.441661  ['wait', 'm', 'gon', 'underwater']         ['minute', 'water', 'as_long_as', 'world']
don't_hold_your_breath 0.397862  ['wait', 'groan', 'sweetie', 'apologize']  ['ncic', 'dissection', 'goldilock', 'rowboat']
```

They are waiting with bated breath to see the results.

They are waiting, watching and whispering with bated breath.

Figure 1: Idiomify suggesting idioms that best describe “wait, excitedly, anxiously, hopefully” altogether.

What is the aim?

Building a collocation-supplemented reverse-dictionary of idioms, namely Idiomify, is the controlling aim of the project. Idiomify attempts to suggest a list of idioms that best describe a given set of words, while supplementing the results with collocations of the idioms. **Figure 1** illustrates this with an example scenario. Say we write *They are waiting excitedly, anxiously and hopefully to see the results* to describe the people in the images above. Although the sentence does get the idea across, we may wonder how we could paraphrase the sentence with an idiom, if there is any. We therefore give a set of three words as the input to Idiomify: *excitedly*, *anxiously* and *hopefully.* Given the input, Idiomify suggests *with bated breath*, *hold one's breath* and *don't hold_your breath* as the idioms that are most likely to encompass the meaning of the three words, of which *with bated breath* is found to be the most appropriate one. We therefore rephrase our sentence to *They are waiting with bated breath to see the results.* From the list of "verb collocates" of the idiom, however, we see that *with bated breath* collocates with the verbs *watch* and *whisper.* Because the people in the pictures are indeed watching something and possibly whispering amongst them, we revise our sentence to *They are waiting, watching and whispering with bated breath.*

What is this report about?

This is a what Idiomify does, and this report aims describe why we need it, how we can build it and what improvements we can make on it. - section 2 - the motivations. (why?) - why build a reverse-dictionary of idioms? - why supplement it with collocations of idioms?
- section 3 - the methods. (how?)
- how do we build a reverse-dictionary of idioms? - related work? - so how are we doing it?
- how do we extract collocations of idioms from corpora?
- related work? - so how are we doing it?

- section 4 - the results and discussions (improvements?)
  - wha

2. Motivations

Why build a reverse-dictionary of idioms?

Why extract the collocations of idioms?

Objectives (restated)

We therefore

Methods and

How

2. Introduction

2.2 Motivation

> Why build a reverse-dictionary?
>
> Why build a reverse-dictionary of idioms?
>
> why extract collocations?
>
> Why extract collocations of idioms? - helps L2 learners use

2.3 Related Work

> How have people dealt with building a reverse-dictionary? - make it chronological.. - a graph-based approach: WordNet. - a machine-learning based appraoch: LSTM + Word2Vec
>
> So am I doing the same way as they did? - We don't have "IdiomNet". Hence, though appealing, graph-based approach is not feasible. - The second approach: might be feasible - but the definitions of idioms are not as abundant as those for words.
>
> Then how are you going to do about it? - why not just build an inverted index? - In order to experiment, to see if we could do this purely with distributional semantics. -> conclusion later: should be used in conjunction with an inverted-index approach. (That's what I'll be doing in the summer, anyways.)

2.4 Objectives

Just list out all of the things that will be discussed in the following, methods section.

---

3. Methods and Justifications

3.1 Identifying Idioms (`identify-idioms`)

> What & why?

It is imperative to build an automatic pipeline for identify idioms - what set of idioms are we targeting at?

How? - the challenge.

It is challenging to identify idioms because they extensively vary in forms. Idioms are classified into two types: those that are syntactically frozen, i.e. non-compositional, and those that are more flexible, i.e. decomposable (Fischer and Keil, 1996). While non-compositional idioms are readily identifiable in general by searching their nearly fixed forms (e.g. *by hook or by crook*), the hyphenated ones of which are not, because we often omit hyphens. For instance, both *down-to-earth* and *down to earth* occur frequently in corpora. Compared with non-compositional idioms, it is considerably more difficult to identify decomposable idioms because their constituent verbs and personal pronouns inflect. For example, *find one's feet* may morph into *found your feet*, *finding my feet*, etc. Different types aside, idioms often occur in alternative forms, e.g. *heap insult on injury* is an alternative for *add insult to injury* (*Wiktionary*, 2021). This, together with optional hyphens and inflecting forms, poses a challenge to identifying idioms.

base form	matching rule
down-to-earth	`[[LEMMA:down];[OPTIONAL:-];[LEMMA:to];[OPTIONAL:-];[LEMMA:`
find one's feet	`[[LEMMA:find];[POS:personal pronoun];[LEMMA:feet]]`
add insult to injury	`[[[LEMMA:add];[LEMMA:insult];[LEMMA:to];[LEMMA:injury]],[`

Table 1: Examples of the matching rules derived from the base form of idioms.

In order to handle their variability as such, we automatically derive matching rules from linguistic features of each idiom. **Table 1** illustrates how the matching rules are derived; Constituent lemmas, hyphens and part-of-speech are detected from the base forms (e.g. *personal pronoun* is derived from *one's*), which are sequentially joined with logical conjunction, denoted as `;`. Alternatives are added to the rules with logical disjunction, denoted as `,`, if they are stated in dictionaries.

Code 2.

idiomatic expression	tokenised
You are ***down to earth****.*	`['you','are','down-to-earth','.']`
Have you ***found your feet*** *on the new job?*	`['have','you',"find one's feet",'on','the','new','job','?']`

idiomatic expression	tokenised

To ask our members to accept a pay cut ***heaps insult on injury****.*	`['to','ask','our','members','to','accept','a','pay','cut',`
	`insult to injury','.']`

Table 2: Examples of the derived matching rules in use.

How? - the solution

The derived matching rules are then used to automatically identify an idiom as an atomic token. **Table 2** exemplifies how the rules identify idioms; In all cases, idioms are identified as an atomic token. If hyphens are omitted, they are corrected to their hyphenated forms (see *down to earth*). Inflectional and alternative variations are detected and normalised to their base forms i.e. lemmatised (see *find one's feet* and *add insult to injury*). Deriving the rules and identifying idioms are done automatically with a python library that is developed by the author (Kim, 2021).

3.2 Pre-processing Data

3.3 Modeling & Extracting Collocation

3.4 Training Idiom2Vec (`idiom2vec`)

3.5 Suggesting idioms with Idiomify (`idiomify`)

4. Results and Discussions

4.1 Idiom2collocations

evaluation?

first, a preliminary experiment. - better get rid of proper nouns - don't render any collocational insights.

Good side - e.g. fair-and square.

Bad side - find an example

How could we improve upon this? - try different measures ... e.g. pearson, t-test. - But the problem is, what model fits the concept of collocation best? - this is a niche for machine learning. (Cite that paper.) - a research has been done about this -

MACHINE LEARNING FOR COLLOCATION IDENTIFICATION (Yang, 20??).

4.2. Idiom2vec

evaluation? - how do I evaluate this. . . ? We evaluate the model with idiomify.

good side - show the results for most_similar_idioms.

bad side & what did you learn? - show the results for most_similar_words of `in_the_zone` - why is that? 1.I've trained it on idiom2sent dataset only. Not on the entire dataset. - **how is this flawed, mathematically speaking?** This is the part you really have to 2. also, a second reason - maybe I shouldn't have lemmatised them. (e.g. in_the_zone -> focused, immersed,)

how could we improve this? - Train word2vec on the entire corpus. - Train with a un-lemmatised corpus.

4.3. Idiomify

evaluation?

good side

bad side & what did you learn?

how could you improve upon it? 1. combination of inverted index search + word2vec. - (though previously, you did this only with word2vec) Provided that you have a set of definitions of idioms (can be solved with)

2. or, for full-machine learning approach: Take a BERT-based approach to this.

- will be able to go beyond bow assumption, and idiomify an actual phrase.
- will need to carefully labeled definitions of idioms.
-

---

5. Conclusion

Basically a summary of all.

- the collocations: the methods proposed could be used to compile a "Learner's Dictionary of Collocations of English Idioms"

6. References