Grasping the Finer Point: A Supervised Similarity Network for Metaphor Detection

(Rei, Bulat, Kiela, Shutova; 2017)

Silvan de Boer

April 2019

Grasping the Finer Point: A Supervised Similarity Network for Metaphor Detection

Marek Rei[♣♠] Luana Bulat[♣] Douwe Kiela[♢] Ekaterina Shutova[♣]
[♣]Computer Laboratory, University of Cambridge, United Kingdom
[♠]The ALTA Institute, University of Cambridge, United Kingdom
[♢]Facebook AI Research, New York, USA
{marek.rei,luana.bulat,ekaterina.shutova}@cl.cam.ac.uk, dkiela@fb.com

Abstract

The ubiquity of metaphor in our everyday communication makes it an important problem for natural language understanding. Yet, the majority of metaphor processing systems to date rely on hand-engineered features and there is still no consensus in the field as to which features are optimal for this task. In this paper, we present the first deep learning architecture designed to capture metaphorical composition. Our results demonstrate that it outperforms the existing approaches in the metaphor identification task.

1 Introduction

Metaphor is pervasive in our everyday communication, enriching it with sophisticated imagery and helping us to reconcile our experience in the world with our conceptual system (Lakoff and Johnson, 1980). In the most influential account of metaphor to date, Lakoff and Johnson explain the phenomenon through the presence of system-

dominantly on classifying linguistic expressions as literal or metaphorical. They experimented with a range of features, including lexical and syntactic information (Hovy et al., 2013; Beigman Klebanov et al., 2016) and higher-level features such as semantic roles (Gedigian et al., 2006), domain types (Dunn, 2013), concreteness (Turney et al., 2011), imageability (Strzalkowski et al., 2013) and WordNet supersenses (Tsvetkov et al., 2014). While reporting promising results, all of these approaches used hand-engineered features and relied on manually-annotated resources to extract them. In order to reduce the reliance on manual annotation, other researchers experimented with sparse distributional features (Shutova et al., 2010; Shutova and Sun, 2013) and dense neural word embeddings (Bracewell et al., 2014; Shutova et al., 2016). Their experiments have demonstrated that corpus-driven lexical representations already encode information about semantic domains needed to learn the patterns of metaphor usage from linguistic data.

We take this intuition a step further and present the first deep learning architecture designed to

Table of Contents

Table of Contents

Metaphor identification

Literal or Metaphorical?

Absorb cost
Digest milk
Leak news
Green energy
Gold coin

Metaphor identification

Literal or Metaphorical?

Absorb cost
Digest milk
Leak news
Green energy
Gold coin

Approaches so far

- Hand-coded lexical knowledge
- Corpus-driven lexical representations

Research questions

Can a deep learning model capture metaphorical composition?

- What model configuration works best?
- How important is the amount of training data?
- How well does the model transform the input space?

Contributions

- Supervised Similarity Network
- State-of-the-Art performance
- Promise of more data

Table of Contents

Inspiration

Black Holes and White Rabbits:
Metaphor Identification with Visual Features

Ekaterina Shutova
Computer Laboratory
University of Cambridge
es407@cam.ac.uk

Douwe Kiela
Computer Laboratory
University of Cambridge
dk427@cam.ac.uk

Jean Maillard
Computer Laboratory
University of Cambridge
jean@maillard.it

Abstract

Metaphor is pervasive in our communication, which makes it an important problem for natural language processing (NLP). Numerous approaches to metaphor processing have thus been proposed, all of which relied on linguistic features and textual data to construct their models. Human metaphor comprehension is, however, known to rely on both our linguistic and perceptual experience, and vision can play a particularly important role when metaphorically projecting imagery across domains. In this paper, we present the first metaphor identification method that simultaneously draws knowledge from linguistic and visual data. Our results demonstrate that it outperforms linguistic and visual models in isolation, as well as being competitive with the best-performing metaphor identification methods, that rely on hand-crafted knowledge about domains and perception.

1 Introduction

Metaphor lends vividness, sophistication and clarity to our thought and communication. At the same time, it plays a fundamental structural role in our cognition, helping us to organise and project knowledge (Lakoff and Johnson, 1980; Feldman, 2006). Metaphors arise due to systematic associations between distinct, and seemingly unrelated, concepts. For instance, when we talk about "the

etc. The existence of this association allows us to transfer knowledge and imagery from the domain of *mechanisms* (the source domain) to that of *political systems* (the target domain). According to Lakoff and Johnson (1980), such metaphorical mappings, or *conceptual metaphors*, form the basis of metaphorical language.

Metaphor is pervasive in our communication, which makes it important for NLP applications dealing with real-world text. A number of approaches to metaphor processing have thus been proposed, using supervised classification (Gedigian et al., 2006; Mohler et al., 2013; Tsvetkov et al., 2013; Hovy et al., 2013; Dunn, 2013a), clustering (Shutova et al., 2010; Shutova and Sun, 2013), vector space models (Shutova et al., 2012; Mohler et al., 2014), lexical resources (Krishnakumaran and Zhu, 2007; Wilks et al., 2013) and web search with lexico-syntactic patterns (Veale and Hao, 2008; Li et al., 2013; Bollegala and Shutova, 2013). So far, these and other metaphor processing works relied on textual data to construct their models. Yet, several experiments indicated that perceptual properties of concepts, such as concreteness and imageability, are important features for metaphor identification (Turney et al., 2011; Neuman et al., 2013; Gandy et al., 2013; Strzalkowski et al., 2013; Tsvetkov et al., 2014). However, all of these methods used manually-annotated linguistic resources to determine these properties (such as the MRC concrete-

Features	Method	P	R	$F1$
Linguistic	WordCos	0.73	0.80	**0.76**
	PhrasCos1	0.43	0.96	0.57
Visual	WordCos	0.50	0.95	0.66
	PhrasCos1	0.60	0.91	**0.73**
Multimodal	WordMid	0.59	0.85	0.70
	PhrasMid	0.54	0.93	0.68
	WordLate	0.69	0.72	0.70
	PhrasLate	0.50	1.00	0.67
	MixLate	0.67	0.96	**0.79**

Table 2: System performance on Tsvetkov et al. test set (TSV-TEST) in terms of precision (P), recall (R) and F-score ($F1$)

PhraseCos1 for both verbs and adjectives by 17-19%. This suggests that linguistic word embeddings already successfully capture domain and compositional information necessary for metaphor identification. In contrast, the visual PhraseCos1 model, when applied in isolation, tends to outperform the visual WordCos model. PhrasCos1 measures to what extent the meaning of the phrase can be composed by simple combination of the representations of individual words. In metaphorical language, however, a meaning transfer takes place and this is no longer the case. Particularly in visual data, where no linguistic conventionality and stylistic effects take place, PhrasCos1 captures this property. For adjectives this trend was more evident than for verbs. The visual PhraseCos1 model, even when applied on its own, attains a high F-score of 0.73 on TSV-TEST, suggesting that concreteness and other visual features are highly informative in identification of adjectival metaphors. This effect was present, though not as pronounced, for verbal metaphors, where the vision-only PhraseCos1 attains an F-score of 0.66.

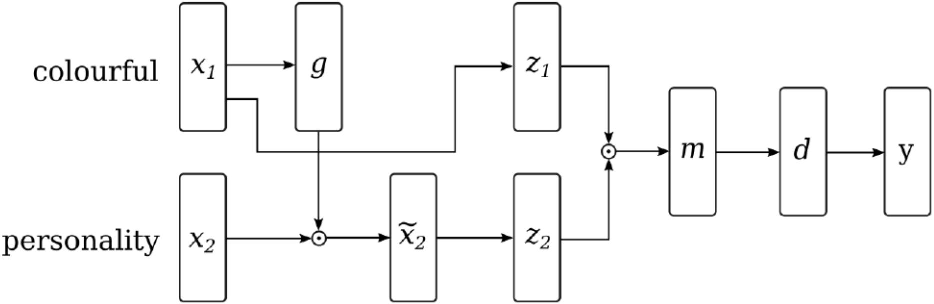
The multimodal model, integrating linguistic and visual embeddings, outperforms the linguistic models for both verbs and adjectives, clearly demonstrating the utility of visual features across word classes. The late fusion method MixLate, which combines the linguistic WordCos score and the visual PhraseCos1, attains an F-score of 0.75 for

embeddings, middle and late fusion techniques attain comparable levels of performance, with WordCos being the leading measure. The reason behind the higher performance of MixLate is likely to be the combination of different scoring methods, one of which is more suitable for the linguistic model and the other for the visual one.

The differences between verbs and adjectives with respect to the utility of visual information can be explained by the following two factors. Firstly, previous psycholinguistic research on abstractness and concreteness (Hill et al., 2014) suggests that humans find it easier to judge the level of concreteness of adjectives and nouns than that of verbs. It is thus possible that visual representations capture the concreteness of adjectives and nouns more accurately than that of verbs. Besides concreteness, it is also likely that perceptual properties in general are more important for the semantics of nouns (e.g. objects) and adjectives (their attributes), than for the semantics of verbs (actions), since the latter are grounded in our motor activity and not merely perception. Secondly, following the majority of multimodal semantic models, we used images as our visual data rather than videos. However, some verbs, e.g. stative verbs and verbs for continuous actions, may be better captured in video than images. We thus expect that using video data along with the images as input to the acquisition of visual embeddings is likely to improve metaphor identification performance for verbal metaphors. However, we leave the investigation of this issue for future work.

In an additional experiment, we evaluated our methods on the larger TSV-TRAIN dataset (specifically using its portion that was not employed for development purposes) and the trends observed were the same. MixLate attained an F-score of 0.71, outperforming language-only and vision-only models. The performance of all scoring methods on TSV-TRAIN was lower than that on the TSV-TEST. This may be the result of the fact that the labelling of TSV-TRAIN was less consistent than that of TSV-TEST. As TSV-TEST is a set of metaphors annotated by 5 annotators with a high agreement, the evaluation on

Model



Training loss

$$E = \sum_{k} q_k$$

$$q_k = \begin{cases} (\tilde{y} - y)^2 & \text{if } |\tilde{y} - y| > 0.4 \\ 0, & \text{otherwise} \end{cases}$$

Word embeddings

- Skip-gram: 100dim
- Attribute-based vectors: 2526dim

SHOES	ANT	DISHWASHER
has_heels, 15	an_insect, 18	an_appliance, 19
has_laces, 13	is_small, 18	requires_soap, 15
worn_on_feet, 13	is_black 15	is_electrical, 14

Data

Mohammad et al.

WordNet Search - 3.1
- WordNet home page - Glossary - Help

Word to search for: absorb [Search WordNet]

Display Options: (Select option to change) [Change]

Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations
Display options for sense: (gloss) "an example sentence"

Verb

- S: (v) **absorb** (become imbued) *"The liquids, light, and gases absorb"*
- S: (v) **absorb**, assimilate, ingest, take in (take up mentally) *"he absorbed the knowledge or beliefs of his tribe"*
- S: (v) **absorb**, take over (take up, as of debts or payments) *"absorb the costs for something"*
- S: (v) **absorb**, suck, imbibe, soak up, sop up, suck up, draw, take in, take up (take in, also metaphorically) *"The sponge absorbs water well"; "She drew strength from the minister's words"*
- S: (v) **absorb** (cause to become one with) *"The sales tax is absorbed into the state income tax"*
- S: (v) **absorb**, take in (suck or take up or in) *"A black star absorbs all matter"*
- S: (v) steep, immerse, engulf, plunge, engross, **absorb**, soak up (devote (oneself) fully to) *"He immersed himself into his studies"*
- S: (v) **absorb** (assimilate or take in) *"The immigrants were quickly absorbed into society"*
- S: (v) **absorb**, engross, engage, occupy (consume all of one's attention or time) *"Her interest in butterflies absorbs her completely"*

Tsvetkov et al.

Most of the rolling hills were sparsely covered with trees

"Please, mark in bold all words that, in your opinion, are used non-literally in the following sentences. In many sentences, all the words may be used literally."

Data

Mohammad et al.

Metaphorical	Literal
absorb cost	accommodate guest
attack problem	attack village
attack cancer	blur vision
breathe life	breathe person
design excuse	deflate mattress
deflate economy	digest milk
leak news	land airplane
swallow anger	swim man

Tsvetkov et al.

Metaphorical	Literal
bloody stupidity	bloody nose
deep understanding	cold weather
empty promise	dry skin
green energy	empty can
healthy balance	frosty morning
hot topix	hot chocolate
muddy thinking	gold coin
ripe age	soft leather

Data

Additional data from Gutierrez et al.

- 23 adjectives, 8.592 phrases

Experiments

- Metaphor identification performance
  - 3 models
  - 2 word embeddings
  - 2 data sets
- Influence of data size on performance
- Qualitative analysis

Table of Contents

Metaphor identification

	Acc	P	R	F1
FNN skip-gram	**76.4**	68.0	**73.8**	**71.0**
FNN attribute	67.7	66.0	70.3	69.4
SSN skip-gram	73.5	**74.3**	73.7	70.2
SSN attribute	72.0	68.1	72.0	65.4
SSN fusion	71.3	68.5	68.1	69.8

Metaphor identification

	Acc	P	R	F1
FNN skip-gram	70.7	67.7	69.7	69.9
FNN attribute	71.9	65.9	75.2	68.3
SSN skip-gram	72.5	73.7	**77.8**	**72.0**
SSN attribute	70.1	68.5	69.3	67.2
SSN fusion	**72.6**	**74.1**	73.9	67.8

Metaphor identification

	Acc	P	R	F1
FNN skip-gram	71.2	70.4	71.8	70.5
FNN attribute	68.5	66.7	71.0	68.3
SSN skip-gram	**74.8**	**73.6**	**76.1**	**74.2**
SSN attribute	69.7	68.8	69.7	68.8
SSN fusion	70.8	70.1	70.9	69.9

Metaphor identification

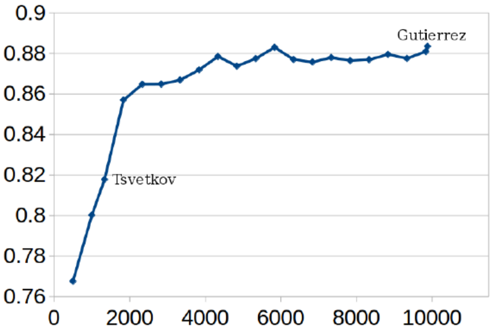
Mohammad et al.

	Acc	P	R	F1
Shutova et al. (2016)				
linguistic	-	67	76	71
multimodal	-	65	**87**	**75**
FFN skip-gram	71.2	70.4	71.8	70.5
FFN attribute	68.5	66.7	71.0	68.3
SSN skip-gram	**74.8**	**73.6**	**76.1**	**74.2**
SSN attribute	69.7	68.8	69.7	68.8
SSN fusion	70.8	70.1	70.9	69.9

Tsvetkov et al.

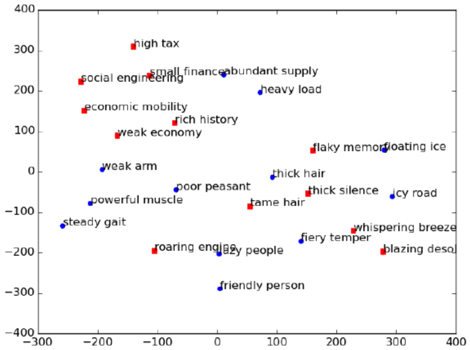
	Acc	P	R	F1
Tsvetkov et al. (2014)	-	-	-	**85**
Shutova et al. (2016)				
linguistic	-	73	80	76
multimodal	-	67	**96**	79
Bulat et al. (2017)	-	85	71	77
FFN skip-gram	77.6	86.6	65.4	74.4
FFN attribute	76.6	82.0	68.6	74.5
SSN skip-gram	82.2	**91.1**	71.6	80.1
SSN attribute	81.9	86.6	**75.7**	80.6
SSN fusion	**82.9**	90.3	73.8	**81.1**

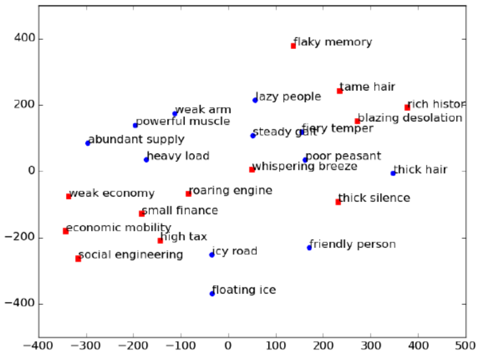
Influence of data size



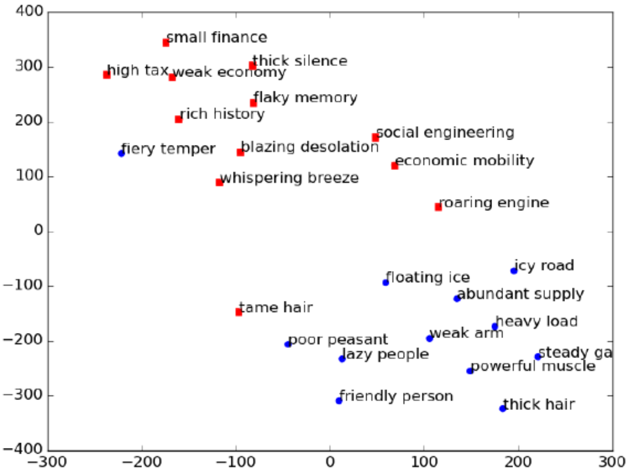
Training data	Acc	P	R	F
Tsvetkov	83.0	88.3	76.3	81.8
Tsvetkov+Gutierrez	**88.7**	**91.6**	**85.4**	**88.3**

Qualitative analysis





Qualitative analysis



Qualitative analysis

Input phrase	Gold	Predicted	Score
sunny country	0	0	0.152
sweet treat	0	0	0.358
lost wallet	0	0	0.439
meaningless discussion	0	0	0.150
gentle soldier	0	0	0.175
unforgiving heights	1	1	0.867
easy money	1	1	0.503
blind hope	1	1	0.813
rolling hills	1	1	0.677
educational gap	1	1	0.827
humane treatment	0	1	0.617
democratic candidate	0	1	0.510
rich programmer	0	1	0.514
fishy offer	1	0	0.290
backward area	1	0	0.161
sweet person	1	0	0.332

Table of Contents

Opinion

- What is the effect of each network component?
- Metrics: what about AUC?
- SSN fusion: what are the two weights?

Ideas for the future

- Multi-task training using unlabeled data
- Extension based on RNN for longer phrases

Neural Metaphor Detection in Context

Ge Gao[1], Eunsol Choi[1], Yejin Choi[1,2], Luke Zettlemoyer[1]

University of Washington[1]
Allen Institute for Artificial Intelligence[2]

UNIVERSITY
OF AMSTERDAM

Metaphors

A figure of speech in which a word (or phrase) is applied to an object or action to which it is not literally applicable.

I'm drowning in assignments these days.

Previous Research

Used SVO triplets

- Shutova et al., 2016
- Tsvetkov et al., 2013
- Rei et al., 2017
- Bulat et al., 2017

When using full sentences, used unigram-based features

- Köper and im Walde, 2017
- Turney et al., 2011
- Jang et al., 2016

The Tasks

Sequence Labelling

Input: sentence $x_1, \ldots, x_n$
Output: binary labels $l_1, \ldots, l_n$
indicating metaphoricity
of each word.

Classification

Input: sentence $x_1, \ldots, x_n$ and a target
verb index i
Output: binary label l indicating
metaphoricity of word x_i

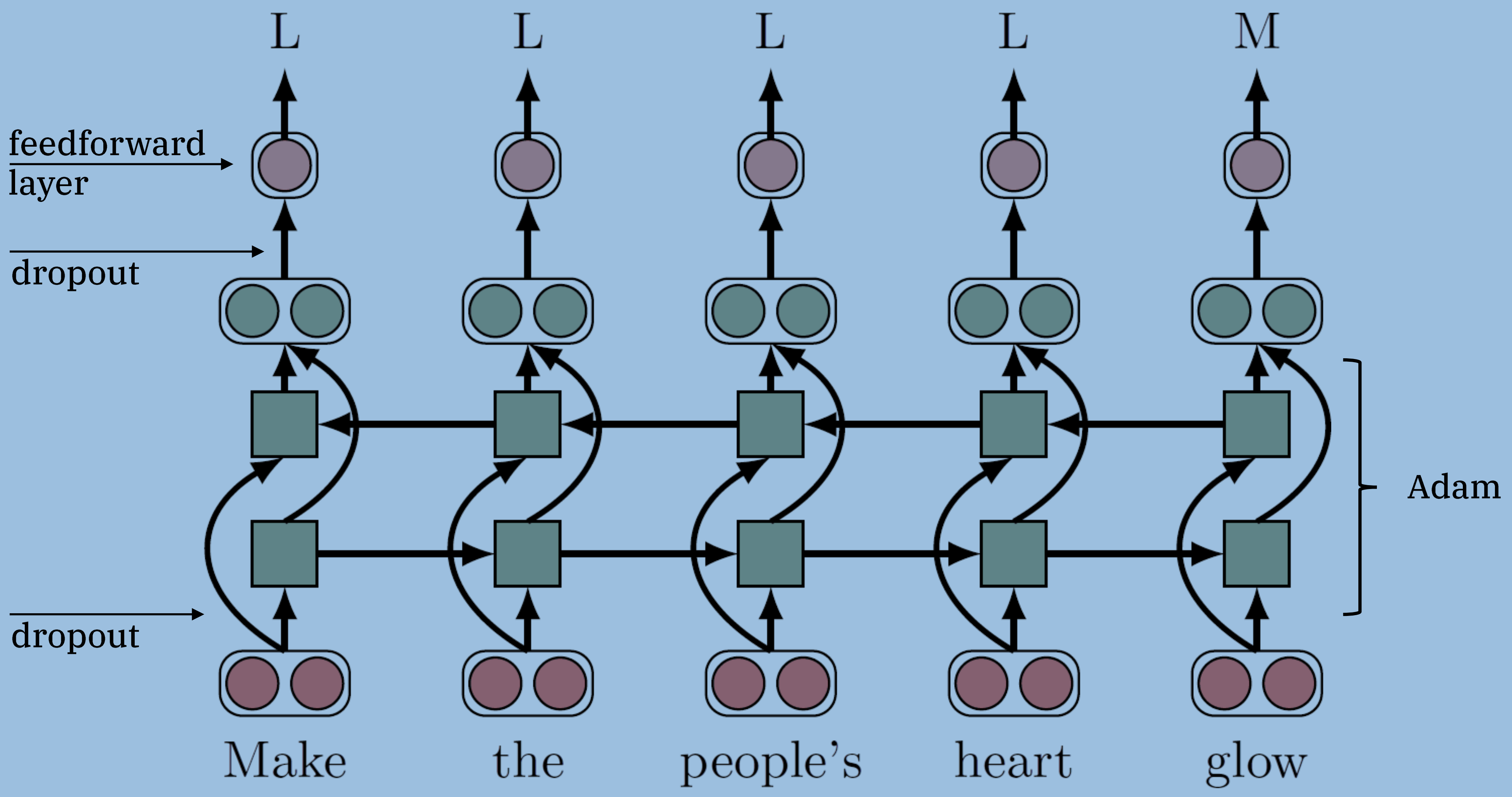
Contribution

End-to-end bi-directional LSTM-based models for metaphor detection, which learn rich contextual word representations useful for the task.

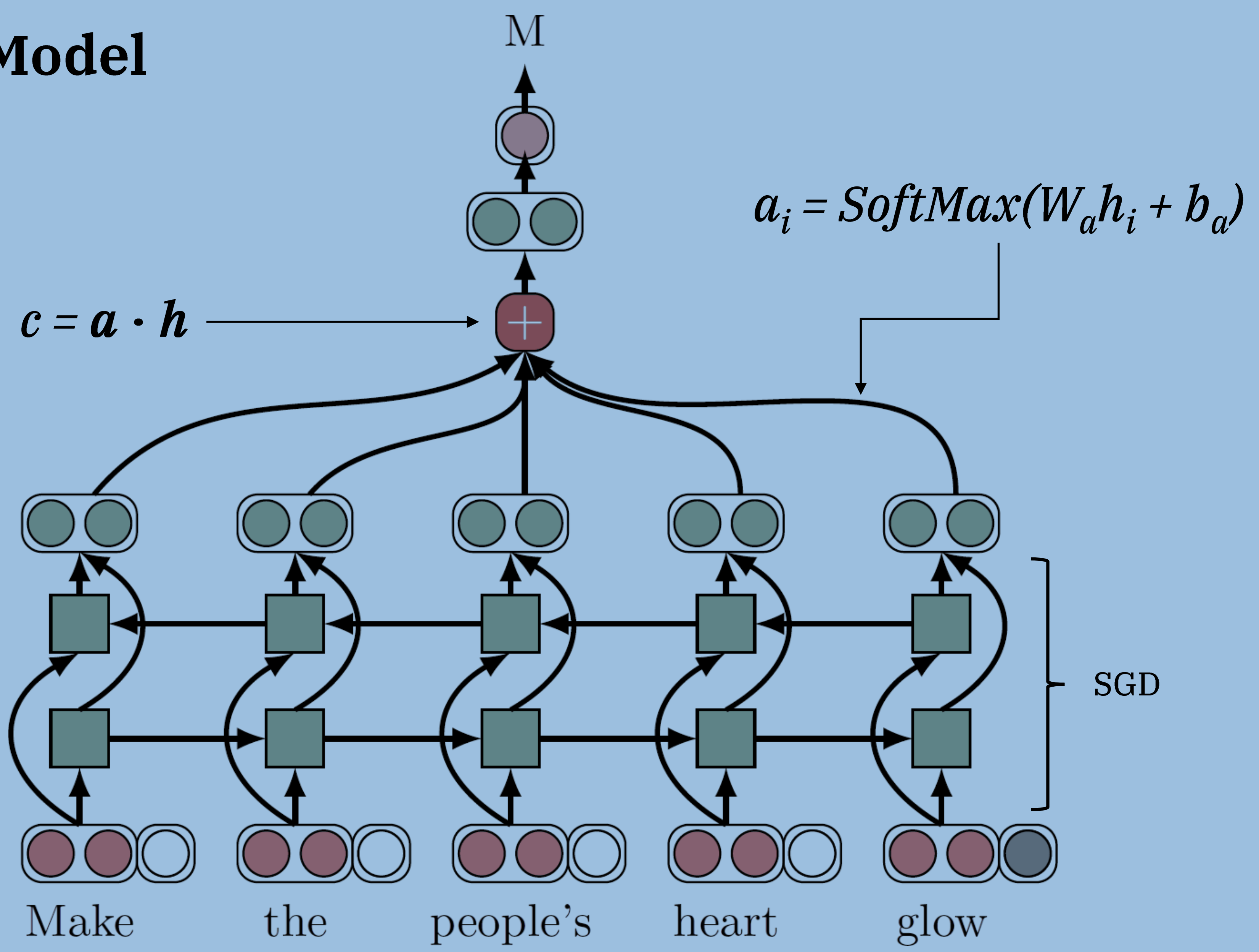
The Input

1. A sentence is tokenized, lemmatized, POS-tagged using **spaCy**
2. Each word is a 300-dimensional **GloVe** embedding
3. Each word embedding is concatenated with the 1024-dimensional **ELMO** embedding
4. (CLS) A 50-dimensional index embedding is appended to each resulting word vector

The SEQ Model



The CLS Model



The Data

	# Expl.	% Metaphor	# Uniq. Verb	Avg # Sent. Len
MOH-X	647	49%	214	8.0
MOH	1,639	25%	440	7.4
TroFi	3,737	43%	50	28.3
VUA	23,113	28%	2047	24.5

Experiments

TroFi, MOH, MOH-X: 10-folds cross validation

VUA: same training, (development), testing set as the VUA verb classification task

Assumption: any unlabelled word is used literally

Results

Model	P	R	F1	Acc.
Lexical Baseline	68.6	45.2	54.5	90.6
Wu (2018) ensemble	60.8	70.0	65.1	-
Ours (SEQ)	**71.6**	**73.6**	**72.6**	**93.1**

Model	MOH-X (10 fold)				TroFi (10 fold)				VUA - Test				
	P	R	F1	Acc.	P	R	F1	Acc.	P	R	F1	Acc.	MaF1
Lexical Baseline	39.1	26.7	31.3	43.6	**72.4**	55.7	62.9	71.4	67.9	40.7	50.9	76.4	48.9
Klebanov (2016)	-	-	-	-	-	-	-	-	-	-	-	-	60.0
Rei (2017)	73.6	76.1	74.2	74.8	-	-	-	-	-	-	-	-	-
Köper (2017)	-	-	-	-	-		**75.0**	-	-	-	62.0	-	-
Wu (2018) ensemble	-	-	-	-	-	-	-	-	60.0	**76.3**	67.2	-	-
CLS	75.3	84.3	**79.1**	**78.5**	68.7	**74.6**	72.0	73.7	53.4	65.6	58.9	69.1	53.4
SEQ	**79.1**	73.5	75.6	77.2	70.7	71.6	71.1	**74.6**	**68.2**	71.3	**69.7**	**81.4**	**66.4**

Discussion

- SEQ model benefits from full sentence annotation in CLS task
- Among false negatives: 50% borderline cases, 33% indirect metaphors, 18% personifications, 2% direct metaphors
- Among false positives: 31% implicit verb arguments, 15% long range dependencies, 10% rare word senses, 5% anthropomorphic arguments

Impact of Elmo

Model	P	R	F1.	Acc.
SEQ	**68.3**	**72.0**	**70.4**	**83.5**
-ELMo	59.4	64.3	61.7	78.2
CLS	52.4	63.0	57.3	74.3
-ELMo	52.0	48.7	50.8	74.1

My Take

- The paper is well written: concise, clear, and to the point
- Unsurprisingly, no statistical tests were done on the results
- Lack of exploration: what about testing (and showing) different strategies for combining hidden states in the CLS model?

  - How are the index embeddings made? Are the input embeddings trained?

Thank You

How to approach your research project

Katia Shutova

ILLC
University of Amsterdam

24 April 2019

Working on a research project

Key steps

1. Formulate your goal or research question
2. Choose methods / models to use
3. Design experiments to test the methods (datasets, baselines)
4. Conduct evaluation: compare the models in terms of performance (quantitative results)
5. Conduct qualitative analysis

Getting started

Project topics come with brief project descriptions on Canvas and some suggested literature

1. read the papers on the topic
2. look at the available datasets
3. find out what the state-of-the-art model is for your task
4. build on top of this state-of-the-art model
   - sometimes there can be several types of models (near-SOTA)
   - numbers alone should be taken with a grain of salt
5. use ideas and models studied in the course, and research wider literature

Designing experiments

1. Choose your baselines wisely:
   - make sure the models are **comparable**
   - a good baseline model does everything the way your model does, except for **the one thing** that you are evaluating
2. Perform ablation experiments:
   - add one technique at a time
   - determine its contribution
3. Compare to prior research (when possible)

Training and evaluation: good research practice

- **Training, development** and **test** splits
  - development set used for parameter tuning
  - test set kept unseen!
  - use standard split, if available in the literature
- **Cross-validation**
  - a viable alternative for smaller datasets
  - use stratification
  - standard dataset splits may be available
- Our friend: **statistical significance**!

Conducting experiments: the reality

You came up with your brilliant idea!

You have performed all of the above steps perfectly!

And yet... it doesn't work...

What do you do next?

Conducting experiments: the reality

You came up with your brilliant idea!

You have performed all of the above steps perfectly!

And yet... it doesn't work...

What do you do next?

Not this...

Also not this...

You do this

- Try to diagnose the problem
  - look at the data, perform **error analysis**
  - play with **parameter settings**
  - conduct an experiment under "ideal conditions":
    e.g. equal dataset sizes in a multitask learning setup
  - also **talk to us** at this point!
- Change your setup and try again
  - experiment with a **different dataset**
  - experiment with **variants of the model**, or a different architecture
- Getting a positive result often requires several iterations!

You do this

- Try to diagnose the problem
  - look at the data, perform **error analysis**
  - play with **parameter settings**
  - conduct an experiment under "ideal conditions":
    e.g. equal dataset sizes in a multitask learning setup
  - also **talk to us** at this point!
- Change your setup and try again
  - experiment with a **different dataset**
  - experiment with **variants of the model**, or a different architecture
- Getting a positive result often requires several iterations!

You do this

- Try to diagnose the problem
  - look at the data, perform **error analysis**
  - play with **parameter settings**
  - conduct an experiment under "ideal conditions":
    e.g. equal dataset sizes in a multitask learning setup
  - also **talk to us** at this point!
- Change your setup and try again
  - experiment with a **different dataset**
  - experiment with **variants of the model**, or a different architecture
- Getting a positive result often requires several iterations!

Conducting an analysis

1. Find ways to visualise different aspects of your model
   - e.g. graphs, tSNE plots etc
2. Investigate model behaviour under different conditions
   - e.g. the effect of training data size
   - or performance across different classes
3. Qualitative analysis
   - perform error analysis
   - what does your model do well and where does it fail
   - other interesting trends that the data shows

Conducting an analysis

1. Find ways to visualise different aspects of your model
   - e.g. graphs, tSNE plots etc
2. Investigate model behaviour under different conditions
   - e.g. the effect of training data size
   - or performance across different classes
3. Qualitative analysis
   - perform error analysis
   - what does your model do well and where does it fail
   - other interesting trends that the data shows

Conducting an analysis

1. Find ways to visualise different aspects of your model
    - e.g. graphs, tSNE plots etc
2. Investigate model behaviour under different conditions
    - e.g. the effect of training data size
    - or performance across different classes
3. Qualitative analysis
    - perform error analysis
    - what does your model do well and where does it fail
    - other interesting trends that the data shows