A Semantic Similarity-Based Perspective of Affect Lexicons for Sentiment Analysis

Oscar Araque
Ganggao Zhu
Carlos A. Iglesias

SIMON: SIMilarity-based sentiment projectiON

Outline

- Introduction
- Background
  - Sentiment Lexicons
  - Knowledge-based similarity: WordNet
  - Corpus-based similarity: embeddings
- Proposed model
  - Semantic similarity
  - Embedding representation
- Evaluation
  - Semantic feature extraction
  - Sentiment classification evaluation
  - Vocabulary selection
- Conclusions and future work

Outline

- **Introduction**
- Background
  - Sentiment Lexicons
  - Knowledge-based similarity: WordNet
  - Corpus-based similarity: embeddings
- Proposed model
  - Semantic similarity
  - Embedding representation
- Evaluation
  - Semantic feature extraction
  - Sentiment classification evaluation
  - Vocabulary selection
- Conclusions and future work

Introduction

- In Sentiment Analysis (SA), **sentiment lexicons** are frequently used
  - Lexicons represent direct subjective sentiment signals
  - A resource that is easy to incorporate to our systems

- Lexicons can be:
  - Manually or automatically generated
  - Of general scope or adapted to a domain
  - Trade-off between coverage and precision

Introduction: Proposal

- We propose to use a sentiment lexicon in a novel way
- Instead of **keyword matching**, we propose a **similarity-based projection** over selected sentiment words
- The representation of a text is done by how similar **input words** are to **lexicon words**

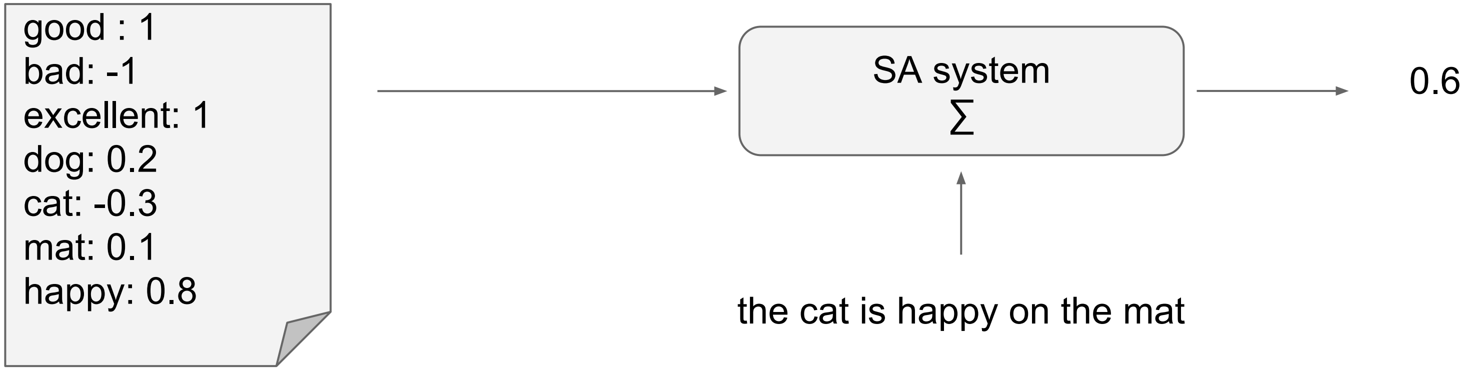
Outline

- Introduction
- **Background**
  - Sentiment Lexicons
  - Knowledge-based similarity: WordNet
  - Corpus-based similarity: embeddings
- Proposed model
  - Semantic similarity
  - Embedding representation
- Evaluation
  - Semantic feature extraction
  - Sentiment classification evaluation
  - Vocabulary selection
- Conclusions and future work

Background: sentiment lexicons

- Typically, lexicons are used through **keyword matching** or Bag-of-words approaches

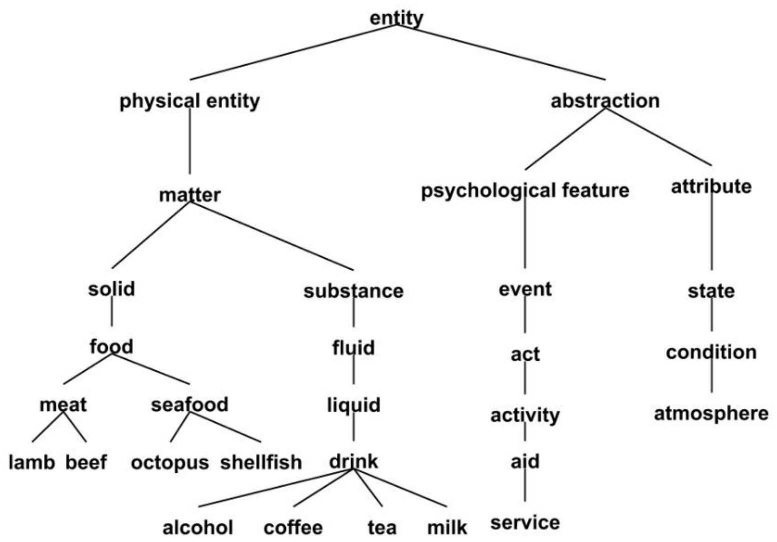
- Keyword-matching example:



Background: Knowledge-based similarity

- These methods measure the similarity using an ontology

- There are different similarity measures, based on the **path** between words in the **taxonomy**



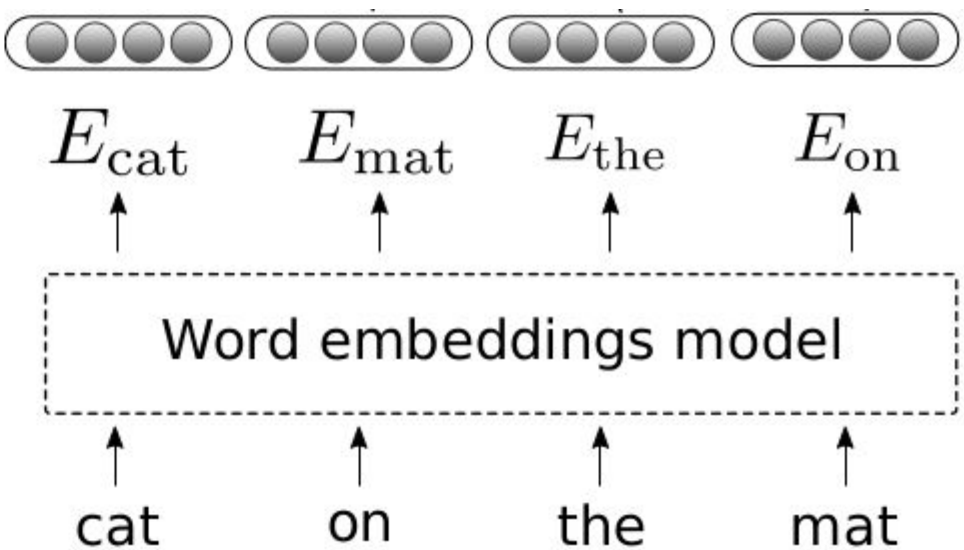
Background: Corpus-based similarity

- Another popular source of information are **word embeddings**
  - Unsupervised training from a lot of text (corpus)
  - Capture language regularities, semantics
    - Many natural language information is contained in embeddings

- Information is captured by word co-occurrence rather that by manual annotation

Background: word embeddings

- An embedding model can be considered as a mapping between a word and a vector in a *n*-dimensional space



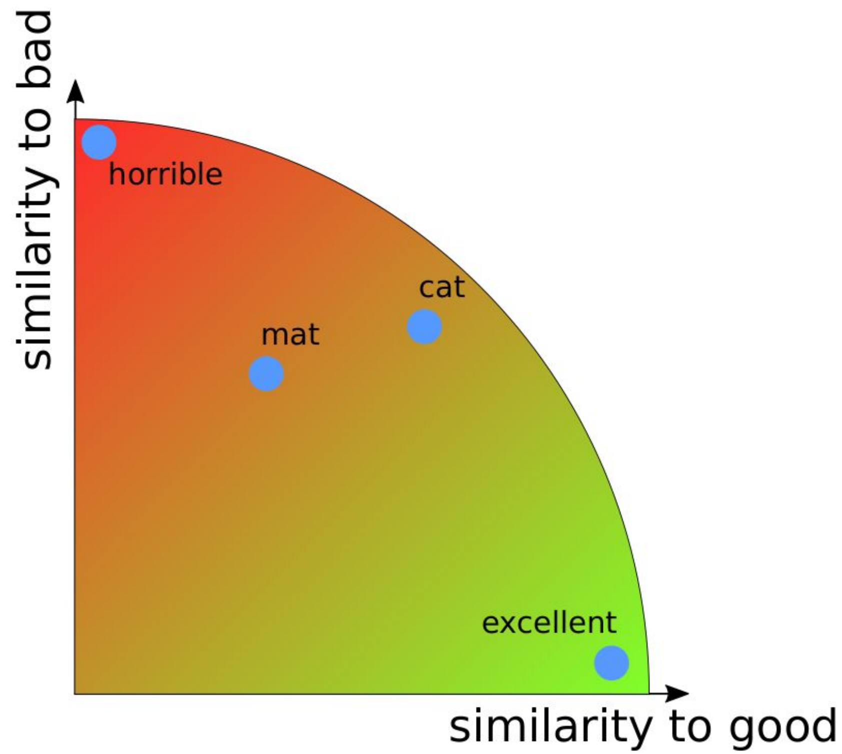
- Similarity between words can be computed using the **dot product**

Outline

- Introduction
- Background
  - Sentiment Lexicons
  - Knowledge-based similarity: WordNet
  - Corpus-based similarity: embeddings
- **Proposed model**
  - Semantic similarity
  - Embedding representation
- Evaluation
  - Semantic feature extraction
  - Sentiment classification evaluation
  - Vocabulary selection
- Conclusions and future work

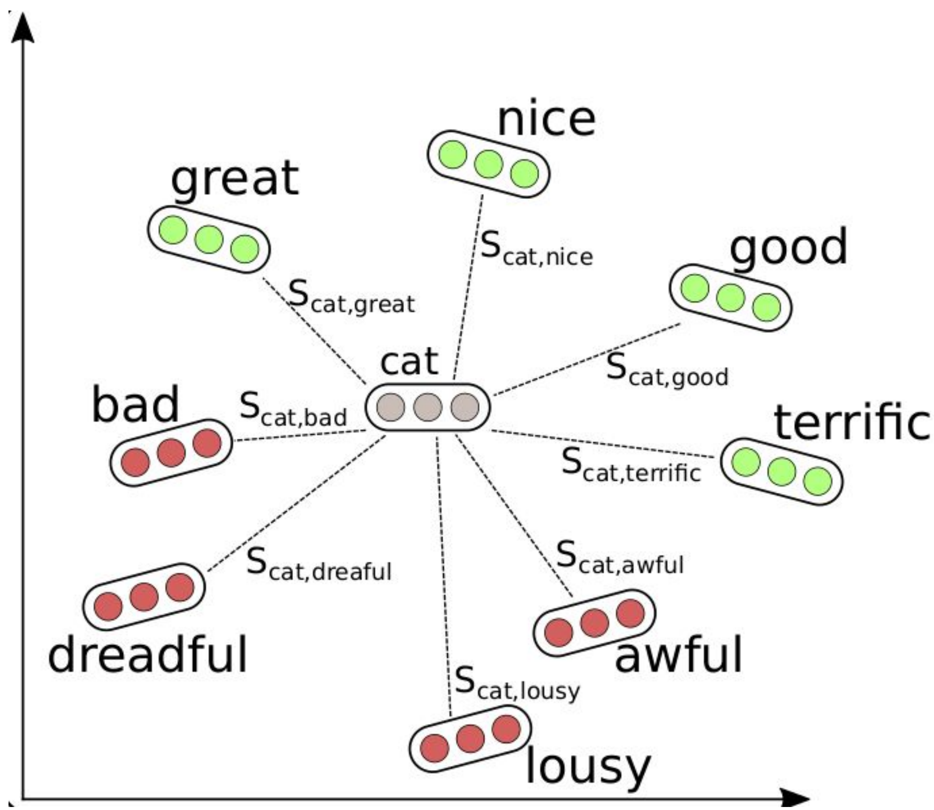
Proposed model: Lexicon words projection

- A word can be represented as a vector with the values of the similarities to a word selection
- E.g., projection over {good, bad}:



Proposed model:
Lexicons words projection

- If we consider a higher number of lexicon words, a 2D visualization could be:



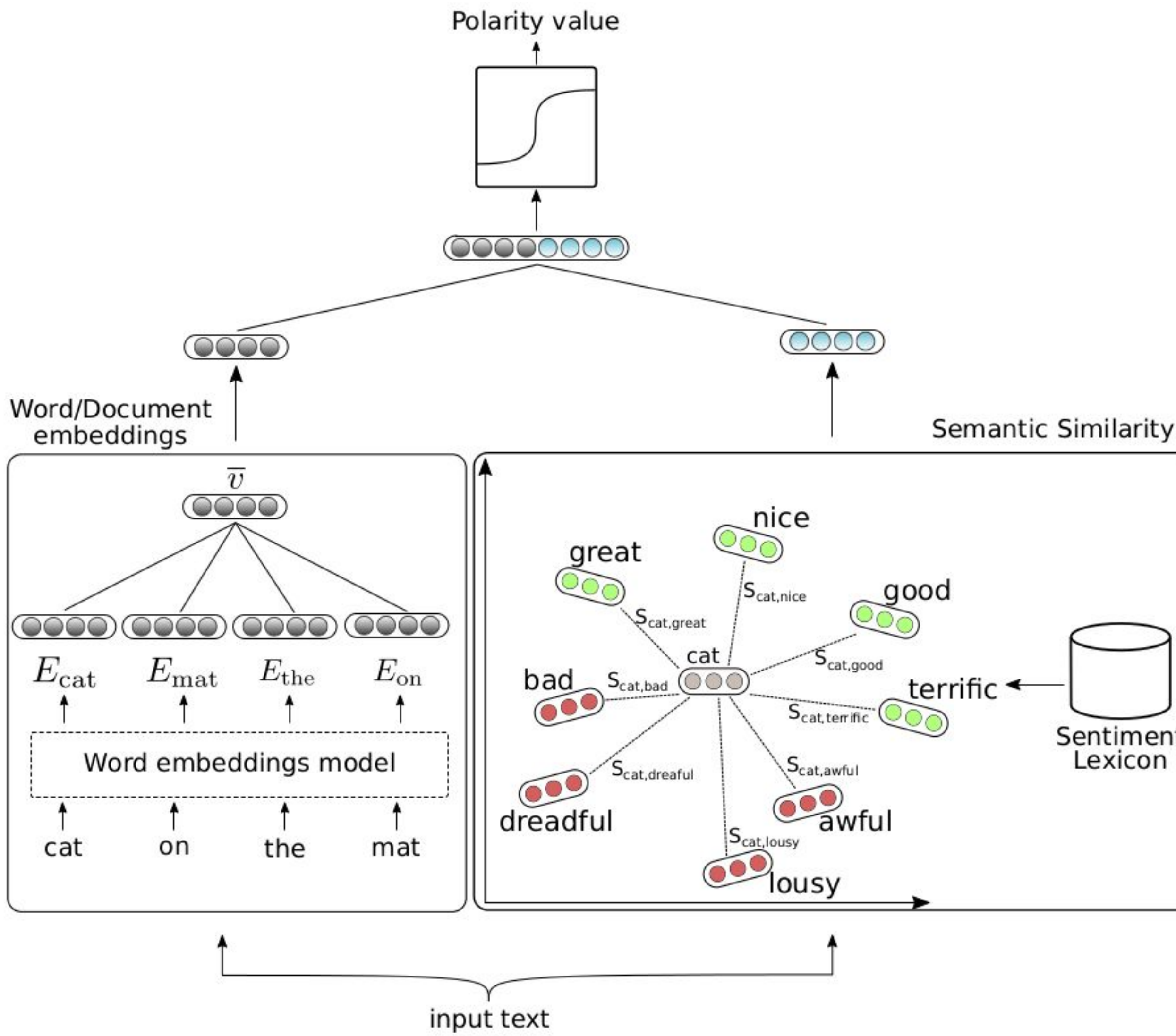
Proposed model: Features computation

	l_1	$\cdots$	l_j	$\cdots$	l_L
w_1	$\mathrm{sim}(w_1, l_1)$		$\mathrm{sim}(w_1, l_j)$		$\mathrm{sim}(w_1, l_L)$
$\vdots$					
w_i	$\mathrm{sim}(w_i, l_1)$		$\mathrm{sim}(w_i, l_j)$		$\mathrm{sim}(w_i, l_L)$
$\vdots$					
w_I	$\mathrm{sim}(w_I, l_1)$		$\mathrm{sim}(w_I, l_j)$		$\mathrm{sim}(w_I, l_L)$

$$\Big\downarrow \text{Pooling}$$

$$\boxed{p_1 \quad \cdots \quad p_j \quad \cdots \quad p_L}$$

Proposed model: architecture

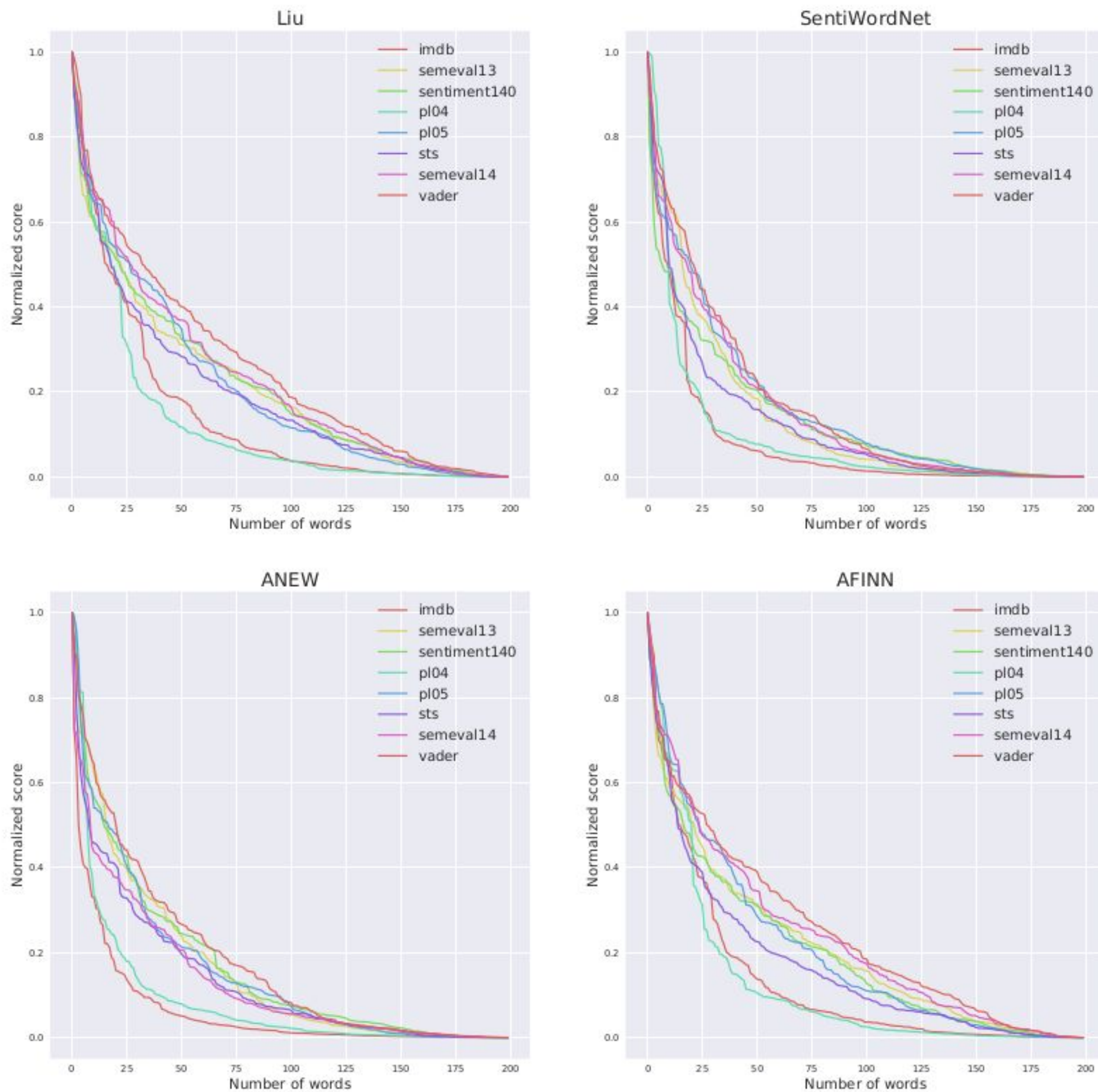


Outline

- Introduction
- Background
  - Sentiment Lexicons
  - Knowledge-based similarity: WordNet
  - Corpus-based similarity: embeddings
- Proposed model
  - Semantic similarity
  - Embedding representation
- **Evaluation**
  - Word selection
  - Sentiment classification evaluation
  - Cross-dataset evaluation
- Conclusions and future work

Evaluation:
Word selection

- Distribution is consistent on all datasets
- Selection over 25% of total words



Evaluation: Sentiment analysis

- 7 datasets and 4 lexicons

- For lexicon word keyword matching:

Dataset	SemEval13	SemEval14	Vader	STS	IMDB	PL04	PL05
Liu	76.53	73.36	80.25	**67.78**	73.49	68.33	**61.98**
SentiWordNet	69.88	68.36	67.21	50.00	66.24	64.73	55.64
ANEW	71.75	69.26	66.43	54.17	66.41	65.96	54.24
AFINN	**80.55**	**78.76**	**87.22**	67.10	**73.58**	**68.90**	60.94

Evaluation: Sentiment analysis

- For embedding-based similarity

Dataset	SemEval13	SemEval14	Vader	STS	IMDB	PL04	Pl05	Rank
W2V/D2V	84.54	84.14	88.02	**83.75**	88.53	88.65	76.43	3.7
Liu	79.61	78.75	85.48	78.69	82.13	84.02	74.15	7.1
Liu + W2V/D2V	**87.09**	**86.48**	**90.39**	82.60	**88.99**	**89.45**	78.25	1.6
SentiWordNet	76.62	74.31	84.77	79.15	81.66	80.11	73.95	8.3
SentiWordNet + W2V/D2V	82.45	81.29	87.66	81.72	88.82	88.03	78.26	4.3
ANEW	79.39	78.75	86.91	76.60	79.42	76.66	74.21	7.8
ANEW + W2V/D2V	86.30	85.65	90.08	77.54	88.88	88.09	**78.29**	3.6
AFINN	81.53	79.17	86.13	80.60	81.99	82.27	74.16	6.4
AFINN + W2V/D2V	86.68	85.92	90.26	83.29	88.97	88.84	78.09	2.3

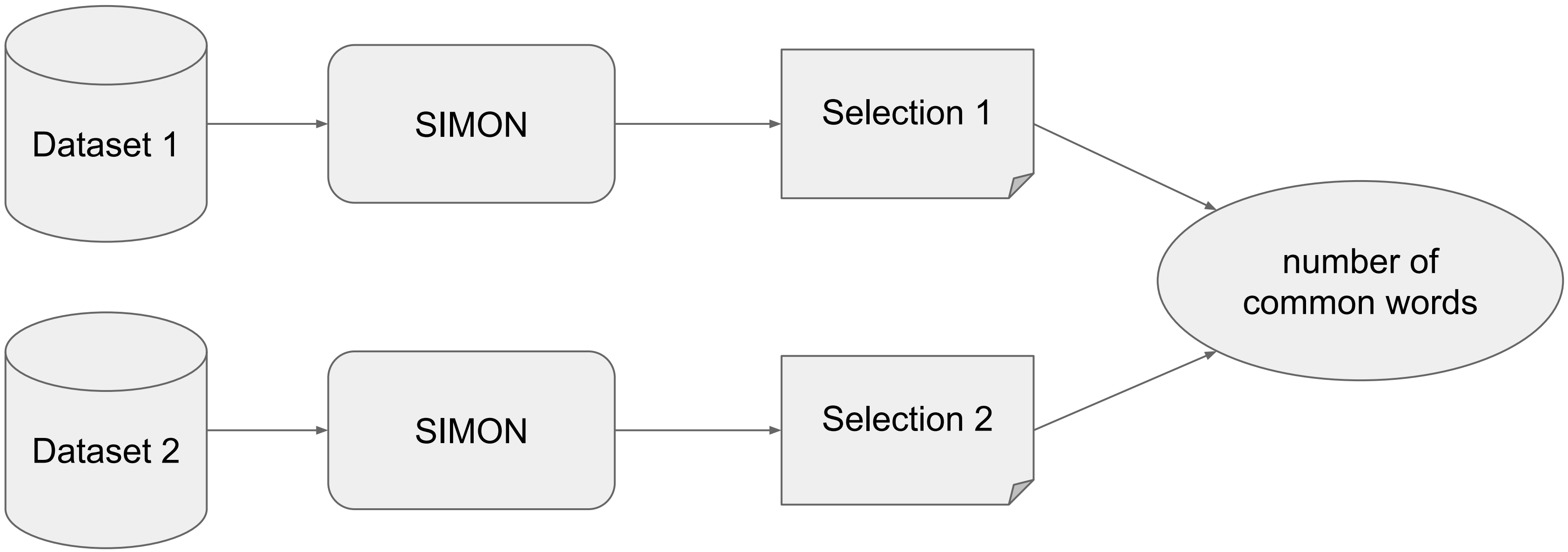
Evaluation: Sentiment analysis

- For WordNet-based similarity (WPath)

Dataset	SemEval13	SemEval14	Vader	STS	IMDB	PL04	Pl05	Rank
W2V/D2V	84.64	**84.11**	88.19	**83.75**	88.55	88.75	76.25	2.7
Liu	56.45	51.55	72.10	62.79	72.90	68.31	57.47	7.7
Liu + W2V/D2V	84.31	83.00	88.44	83.07	88.51	88.87	75.98	4.1
SentiWordNet	66.49	61.18	73.23	62.68	71.47	72.23	56.19	7.4
SentiWordNet + W2V/D2V	85.06	83.75	88.22	83.29	88.52	88.78	**76.39**	2.7
ANEW	63.11	54.84	71.88	59.75	71.50	68.50	56.61	8.3
ANEW + W2V/D2V	84.31	83.08	87.58	83.58	**88.58**	88.85	75.99	3.5
AFINN	66.97	58.56	75.80	66.28	72.61	69.05	56.76	6.6
AFINN + W2V/D2V	**84.83**	83.25	**88.95**	83.64	88.54	**89.23**	76.26	2

Evaluation:
Cross-dataset experiments

- If performing in a new dataset, how can we ***predict*** the performance of our method?
- We define the **number of common words** between two sets of selected words



Evaluation: Cross-dataset experiments

- Performing a Least Square analysis between:
  - Difference of performance across two datasets
  - Number of common words across two word selections

- R^2 values are:
  - Liu: 0.93 ($p < 0.01$)
  - SentiWordNet: 0.94 ($p < 0.01$)
  - ANEW: 0.92 ($p < 0.01$)
  - AFINN: 0.89 ($p < 0.01$)

- Using this metric, the difference in performance in a new dataset can be estimated

Evaluation: Additional explorations

- The idea to the check is similar lexicons yield similar performances using SIMON
- We have defined **4 metrics** that measure the similarity/distance between two lexi
- Pearson correlation - r = -0.71 ($p < 0.01$)
  - Distance between two selection of words
  - Cross-dataset performance
- We can have a sense of the performance of a new lexicon by the distance to another studied lexicons

Outline

- Introduction
- Background
  - Sentiment Lexicons
  - Knowledge-based similarity: WordNet
  - Corpus-based similarity: embeddings
- Proposed model
  - Semantic similarity
  - Embedding representation
- Evaluation
  - Word selection
  - Sentiment classification evaluation
  - Cross-dataset evaluation
- **Conclusions and future work**

Conclusions

- Proposed SIMON feature extractor is useful for SA
  - Stronger than word-matching approach
- In comparison, embeddings similarity is stronger to WordNet
- It is possible to have a sense of performance over new datasets/lexicons

Current and Future work

- SIMON has been successfully used in other tasks
  - Insomnia detection in Twitter
  - Categorization of radical texts

- SIMON can be extended to work in
  - Emotion analysis
  - A multilingual environment
  - Detection of radicalism

Implementation

- SIMON is integrated in **gsitk**
- Can be used in a scikit-learn Pipeline, which facilitates replication

Thanks!

Oscar Araque
o.araque@upm.es