Acquiring and Exploiting Lexical Knowledge for Twitter Sentiment Analysis

Felipe Bravo-Marquez
Chief Supervisor: Bernhard Pfahringer
Supervisor: Eibe Frank

Department of Computer Science, University of Waikato

21 September,2016

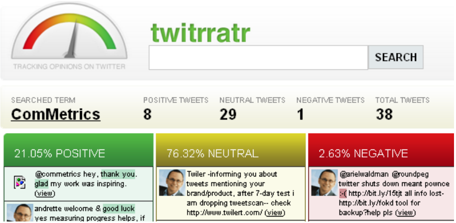
../../img/waikato.png

Social Media

- Microblogging services are increasingly being adopted by people in order to access and publish information.
- **Twitter**: Massively used Microblogging platform where users post messages (a.k.a **tweets**) limited to 140 characters.
- Tweets use a unique **informal dialect** including many abbreviations, acronyms, misspelled words, hashtags, and emoticons, e.g., **lol**, **omg**, **hahaha**, **#hatemonday**, **#SweetAsBro**, **#yeahnah**, **:)** .

Sentiment Analysis and Social Media

- Twitter users tend to publish **personal opinions** regarding certain topics and news events.
    1. Hey @Apple, pretty much all your products are amazing. You blow minds every time you launch a new gizmo. That said, your hold music is crap.
    2. #windows sucks... I want #imac so bad!!! why is it so damn expensive :(@apple please give me free imac and I will love you :D
- Analysing the sentiment underlying these opinions has important applications in product **marketing** and **politics**.



Opinion Mining or Sentiment Analysis

- Application of **NLP** and **text mining** techniques to identify and extract subjective information from textual datasets.

Main Problem: Message-level Polarity Classification (MPC)

1. Automatically classify a tweet to classes **positive**, **negative**, or **neutral**.

2. State-of-the-art solutions use **supervised** machine learning models trained from **manually** annotated examples [**?**].

Drawbacks of Supervised models for MPC

- **Label sparsity (LS)**: manual annotation is **labour-intensive** and **time-consuming**.
- **Concept drift**: the sentiment pattern can vary from one collection to another (domain-drift, temporal-drift).

A classifier trained from tweets annotated for one domain will **not necessarily** work on another one!

Examples of domain-Drift

1. For me the queue was pretty **small** and it was only a 20 minute wait I think but was so worth it!!! :D @raynwise
2. Odd spatiality in Stuttgart. Hotel room is so **small** I can barely turn around but surroundings are inhumanly vast & long under construction.

Solutions to MPC with LS

Using Prior Lexical Knowledge

- An opinion lexicon is a lists of terms labelled by sentiment.
- They are normally composed of positive and negative words such as **happy, wonderful** and **sad, bad**.
- Can be used for **unsupervised** sentiment classification [**?**], or as **low-dimensional** features [**?**].
- Informal Twitter words are **not** covered by most popular lexicons.
- The manual creation of a Twitter-oriented opinion lexicon is a **time-consuming** task.

Solutions to MPC with LS (2)

Distant Supervision

- Automatically **label** unlabelled data (**Twitter API**) using a heuristic method [**?**].
- **Emoticon-Annotation Approach (EAA)**: tweets with positive **:)** or negative **:(** emoticons are labelled according to the polarity indicated by the emoticon [**?**].
- The emoticon is **removed** from the content.
- Drawback: emoticons are **rarely** used in certain domains such as politics.

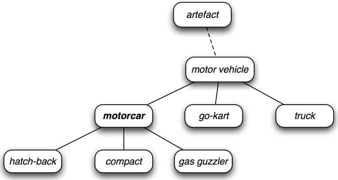
Research Problem

This thesis addresses the label sparsity problem for Twitter sentiment classification by automatically building **two type of resources**.

1. **Twitter-specific opinion lexicons**: we develop machine learning models to induce polarity lexicons from tweets.
2. **Synthetically labelled tweets**: we develop distant supervision methods based on **lexical knowledge** (we go beyond emoticons).

Previous work on Polarity Lexicon Induction (PLI)

- The acquisition is normally done by exploiting **relations** between a **small seed lexicon** and **unknown words** from a **knowledge** resource.
- Two type of resources can be used: a **semantic network** (structured) such as **WordNet**, or a **corpus of documents** (unstructured).



PLI using Semantic Networks

- Methods based on WordNet expand the seed words using semantic relations such **synonyms** and **antonyms**, [**?**, **?**].
- Hypothesis: synonyms have the **same** polarity and antonyms have the **opposite**.
- In [**?**] a **graph** was created using WordNet **adjectives** as vertices and the **synonym** relation as edges.
- Words are expanded by its **relative distance** from the two seed terms **good** and **bad**.
- In [**?**, **?**] the authors take the **dictionary definitions** of the seed words to train a word-level classifier.
- As semantic networks cover a fixed vocabulary, they **cannot** capture informal Twitter words.

Corpus-based PLI

- Corpus approaches exploit **statistical patterns** observed in document corpora (e.g.**tweets**).
- Turney et.al proposed an unsupervised measure called **PMI semantic orientation** (PMI-SO) [**?**].
- It is calculated as the difference between the point-wise mutual information **PMI** of the word with a positive and a negative variable y.

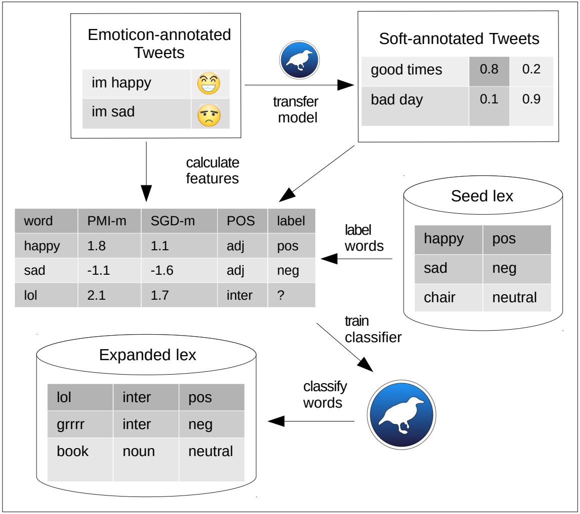
$$\text{PMI-SO}(w) = log_2 \left(\frac{\text{count}(w \wedge y = pos) \times \text{count}(y = neg)}{\text{count}(y = pos) \times \text{count}(w \wedge y = neg)} \right)$$

- For Twitter PLI these variables correspond to message-level labels obtained using **EAA** [**?**, **?**]
- The approach suffers from the same limitations as **EAA** for MPC.

Word-sentiment Associations for PLI

- We propose a **supervised framework** for **PLI** based on **sentiment-annotated tweets**.
- The tweets are labelled in an **automatic fashion** (EAA and transfer model).
- Each expanded word has a **probability distribution**, describing how positive, negative, and neutral it is.
- All the entries of the lexicon are associated with a corresponding **part-of-speech** tag.
- This is useful for word **disambiguation** e.g., fine can be an adjective or a noun.
- These properties are inspired by **SentiWordnet**.

Methodology



Ground-Truth word polarities

- The expansion requires a **seed lexicon** with words labelled by sentiment.
- We create a meta-lexicon by taking the **union** of existing hand-made lexicons.
- We discard all words where a **polarity clash** is observed.

	Positive	Negative	Neutral
AFINN	564	964	0
Bing Liu	2003	4782	0
MPQA	2295	4148	424
NRC-Emo	2312	3324	7714
Seed Lexicon	3730	6368	7088

Table: Lexicon Statistics

Word-level Features

- We prepend a **POS-tag** prefix to each word in order to differentiate **homographs** exhibiting different POS-tags and use the POS tag as a nominal feature.
- We calculate two types of associations for each word: **Stochastic Gradient Descent** (SGD-SO), and **PMI Semantic Orientation** (PMI-SO).
- We calculate associations from **hard** and **soft** message-level labels.

The SGD-SO association

- This SGD-SO association is calculated by incrementally training a **linear support vector machine** from the collection of **hard-labelled** tweets.
- We use **stochastic gradient descent** (SGD) online learning process.

$$\frac{\lambda}{2}||w||^2 + \sum[1 - y(\mathbf{xw} + b)]_+. \tag{1}$$

- We use a squared loss function over the log odds $z = \log_2(\frac{pos(d)}{neg(d)})$ for **soft-annotated** tweets.

$$\frac{\lambda}{2}||w||^2 + \sum(z - (\mathbf{xw} + b))^2. \tag{2}$$

The PMI-SO association

- The second association for **hard-annotated** tweets corresponds to the **PMI semantic orientation** (PMI-SO).

$$\text{PMI-SO}(w) = log_2\left(\frac{\text{count}(w \wedge y = 1) \times \text{count}(y = -1)}{\text{count}(y = 1) \times \text{count}(w \wedge y = -1)}\right) \tag{3}$$

- For soft-annotated tweets:

$$\text{PMI-SO}'(w) = log_2\left(\frac{\sum_{d \in C(w)} \text{pos}(d) \times \sum_{d \in C} \text{neg}(d)}{\sum_{d \in C} \text{pos}(d) \times \sum_{d \in C(w)} \text{neg}(d)}\right) \tag{4}$$

Feature Visualisation

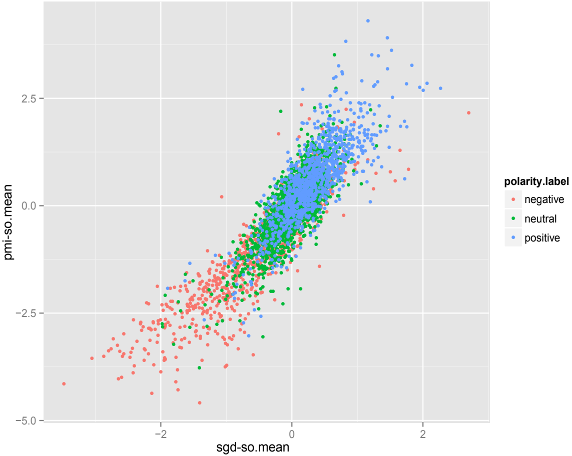


Figure: PMI-SO vs SGD-SO scatterplot.

Word-level Classification Results using RBF SVMs

Weighted AUC		
Dataset	PMI-SO	ALL FEATURES
ED.EM	0.62 ± 0.02	**0.65** $\pm$ **0.02** $+$
STS	0.64 ± 0.02	**0.66** $\pm$ **0.01** $+$
ED.SL	0.63 ± 0.02	**0.65** $\pm$ **0.02** $+$

Table: World-level classification performance.

Expanded Lexicon

(a) (b)

Figure: Word clouds of positive and negative words using log odds proportions.

Message-level classification

	AUC		
Dataset	6HumanCoded	Sanders	SemEval
Seed.Lex	0.77 ± 0.03 = + -	0.77 ± 0.04 = + =	0.77 ± 0.02 = + -
SW	0.74 ± 0.03 - = -	0.7 ± 0.05 - = -	0.76 ± 0.02 = = -
SS	0.81 ± 0.02 + + =	0.78 ± 0.03 = + =	0.81 ± 0.02 + + =
STS	0.82 ± 0.02 + + =	**0.84** ± 0.04 + + +	**0.83** ± 0.02 + + +
ED.EM	0.82 ± 0.03 + + =	0.83 ± 0.04 + + +	0.81 ± 0.02 + + =
ED.SL	0.81 ± 0.02 + + =	0.83 ± 0.04 + + +	0.81 ± 0.02 + + =
ENS	**0.83** ± 0.02 + + =	**0.84** ± 0.04 + + +	**0.83** ± 0.02 + + +

Table: Message-level polarity classification performance. Best result per column is given in bold.

PLI from unlabelled Tweets

- We propose another supervised model for lexicon expansion referred to as the **tweet-centroid model (TCM)**.
- The words are represented by **high-dimensional vectors** based on the context's where they occur.
- In contrast to the previous approach the expansion is done from **unlabelled tweets**.
- It is inspired by the **Distributional Hypothesis** [**?**]: words occurring in the same **contexts** tend to have similar meanings.
- Or equivalently: "a word is characterized by the **company** it keeps".

The Tweet Centroid Model (TCM)

- We treat a **whole tweet** as a word's context.
- We model tweets as **vectors** using standard NLP features.
- We use high-dimensional **unigrams** $\overrightarrow{tb}$ and low-dimensional **word-clusters** $\overrightarrow{tc}$ to form the feature space.
- The word cluster are trained from a corpus of tweets using the **Brown clustering** algorithm [**?**].

The Tweet Centroid Model (TCM) (2)

- The **word-tweet set** $\mathcal{M}(w)$ is the set of tweets from a corpus $\mathcal{C}_U$ in which the word w is observed (posting list in IR):

$$\mathcal{M}(w) = \{m : w \in m\} \tag{5}$$

- The TCM word vector $\overrightarrow{w}$ is the **centroid** of all tweet vectors in $\mathcal{M}(w)$.

$$w_j = \sum_{t \in \mathcal{M}(w)} \frac{x_j^{(t)}}{|\mathcal{M}(w)|} \tag{6}$$

Tweet-centroid Model

sigirmodel.pdf

Datasets

Dataset	STS	ED
#tweets	1, 600, 000	2, 500, 000
#positive words	2015	2639
#negative words	2621	3642
#neutral words	3935	5085
#unlabelled words	36, 451	67, 692
#unigram attributes	45, 022	79, 058
#word-clusters attributes	993	999

Word-level 3-class polarity classification performance

AUC			
Dataset	Unigrams	Brown Clusters	Concatenation
STS	0.77 ± 0.01	$\mathbf{0.79} \pm \mathbf{0.01}$ +	$\mathbf{0.79} \pm \mathbf{0.01}$ +
ED	0.78 ± 0.01	0.79 ± 0.01 +	$\mathbf{0.80} \pm \mathbf{0.01}$ +

Message-level classification performance

AUC			
Dataset	Baseline	STS	ED
Sanders	0.78 ± 0.04	$0.80 \pm 0.04 +$	$\mathbf{0.83} \pm \mathbf{0.04} +$
6-human	0.79 ± 0.03	$0.82 \pm 0.03 +$	$\mathbf{0.83} \pm \mathbf{0.02} +$
SemEval	0.78 ± 0.02	$0.82 \pm 0.02 +$	$\mathbf{0.84} \pm \mathbf{0.02} +$

Other Applications of TCM

- Multi-Label Classification of Emotions.

Other Applications of TCM

Transfer Learning for PLI

- What if we don't have a seed lexicon?
- We can train a **message-level classifier** f_M from a corpus of sentiment annotated tweets $\mathcal{C}_L$ and deploy it on words found in a **corpus of unlabelled tweets** represented by tweet centroids.
- Tweets are represented by **sparse vectors** using unigrams, Brown clusters, and POS tags.
- Note that tweets and words reside in the **same feature space**.

AUC		
Source Dataset	PMI-SO	TCM
Sanders	0.757	**0.864**
6HumanCoded	0.861	**0.930**
SemEval	0.858	**0.916**

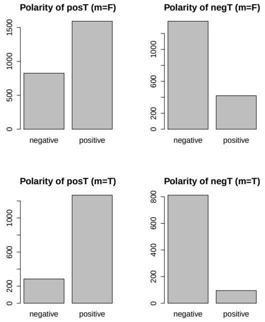
Table: Word-level Polarity Classification Results for the AFINN lexicon.

Lexical-based Distant Supervision

- Lexicons showed to be **useful features** for MPC.
- But we **still need labelled tweets** for training a message-level classifier.
- We will try to **directly use** lexical knowledge for training message-level classifiers.
- We propose two **distant supervision** models: **Partitioned Tweet Centroids** and **Annotate-Sample-Average (ASA)**.
- Proposed methods generate positive and negative training instances by **averaging** tweets containing words with the **same** polarity.

Lexical Polarity Hypothesis

- A tweet containing a word with a certain polarity is more likely to express the **same polarity** than the **opposite** $p_d > 0.5$ (Bernoulli experiment).



- The opposite polarity may also be expressed due to the presence of **negation**, **sarcasm**, or other opinion words with the **opposite** polarity.

Why Averaging?

- Averaging multiple tweets with words with the same polarity **increases** the confidence of generating instances located in the **region** of the desired polarity.
- We assume that the average tweet will behave similarly to the **majority**.
- Probability that the **majority** of the tweets sampled from a collection of tweets with at least one word with the target polarity have the desired polarity:

$$P(M) = \sum_{i=\lfloor \frac{a}{2} \rfloor + 1}^{a} \binom{a}{i} p_d^i (1 - p_d)^{a-i}$$

	$p_d = 0.6$	$p_d = 0.7$	$p_d = 0.8$	$p_d = 0.9$
$a = 3$	0.648	0.784	0.896	0.972
$a = 5$	0.683	0.837	0.942	0.991
$a = 10$	0.633	0.850	0.967	0.998
$a = 50$	0.902	0.998	1	1
$a = 100$	0.973	1	1	1
$a = 500$	1	1	1	1
$a = 1000$	1	1	1	1

- $P(M) > p_d$, when $a \geq 3$ and $p_d \geq 0.5$. This is analogous to the **Condorcet's Jury Theorem**!!

TCM for message-level classification

- TCM can be used as a **distant supervision** model for MPC.
- We use a **word-level** classifier f_W trained with TCM vectors calculated from $\mathcal{C}_U$ labelled by a **polarity lexicon** $\mathcal{L}$ (AFINN).
- The classifier is deployed on the target tweets represented by **sparse vectors**.
- The number of labelled words for training f_W is **limited** to the number of words from $\mathcal{L}$.
- TCM is **not capable** of exploiting large collections of unlabelled tweets for producing training datasets larger than the size of $\mathcal{L}$.

Partitioned TCM

- We propose a modification of our method for **increasing** the number labelled instances it produces.
- The word-tweet set $\mathcal{M}(w)$ for each word from the lexicon ($w \in \mathcal{L}$) is **partitioned** into smaller disjoint subsets $\mathcal{M}(w)_1, \ldots \mathcal{M}(w)_z$ of a fixed size determined by a parameter p.
- We calculate one tweet centroid vector $\overrightarrow{w}$ for **each partition** labelled according to $\mathcal{L}$.

Baselines

Emoticon-Annotation Approach (EAA)

- Labels tweets with positive or negative emoticons according to the emoticon's polarity after removing the emoticon from the message.
- Tweets containing both positive and negative emoticons are **discarded**.

Lexicon-annotation approach (LAA)

- Uses a given polarity lexicon $\mathcal{L}$.
- Tweets with at least one positive word and no negative word are labelled **positive**.
- Tweets with at least one negative word and no positive word are labelled **negative**.

Instances Generated by Distant Supervision Models

We use 10 collections of 2 million tweets as source corpora.

	Avg. Positive	(%)	Avg. Negative	(%)	Avg. Total	(%)
EAA	130,641	(6.5%)	21,537	(1.1%)	152,179	(7.6%)
LAA	681,531	(34.1%)	294,177	(14.7%)	975,708	(48.8%)
TCM	1537	(0.05%)	951	(0.08%)	2488	(0.12%)
TCM (*p*=5)	276,696	(13.8%)	149,989	(7.5%)	426,684	(21.3%)
TCM (*p*=10)	138,596	(6.9%)	75,390	(3.8%)	213,986	(10.7%)
TCM (*p*=20)	69,518	(3.5%)	38,044	(1.9%)	107,563	(5.4%)
TCM (*p*=50)	32,231	(1.6%)	17,950	(0.9%)	50,181	(2.5%)
TCM (*p*=100)	14,338	(0.7%)	8357	(0.4%)	22,695	(1.1%)

TCM for MPC

	6HumanCoded		Sanders		SemEval	
EAA	0.805 ± 0.005	= -	0.800 ± 0.017	= +	0.802 ± 0.006	= -
LAA	0.809 ± 0.001	+ =	0.778 ± 0.002	- =	0.814 ± 0.000	+ =
TCM	0.776 ± 0.004	- -	0.682 ± 0.024	- -	0.779 ± 0.008	- -
TCM (p=5)	0.834 ± 0.002	+ +	0.807 ± 0.008	= +	0.833 ± 0.002	+ +
TCM (p=10)	0.845 ± 0.003	+ +	**0.817** ± 0.006	+ +	0.841 ± 0.002	+ +
TCM (p=20)	**0.850** ± 0.003	+ +	0.815 ± 0.011	+ +	**0.844** ± 0.003	+ +
TCM (p=50)	0.844 ± 0.004	+ +	0.785 ± 0.010	- +	0.836 ± 0.004	+ +
TCM (p=100)	0.829 ± 0.003	+ +	0.752 ± 0.019	- -	0.821 ± 0.004	+ +

Table: Message-level Polarity Classification Results. Best results per column are given in bold.

Annotate-Sample-Average (ASA)

- Partitioned TCM can generate **very large** training datasets.
- TCM instances are obtained by averaging tweets containing **the same word**.
- What if we average random tweets containing **different words** with the same polarity?
- What if we can define the **number of instances** to generate?
- This could be useful for creating **compact and balanced** training datasets.

Annotate-Sample-Average (ASA)

- **Annotation**: every time a word from $\mathcal{L}$ is found, the tweet is added to sets **posT** or **negT** (depending on the polarity).
- Tweets with both positive and negative words will be discarded if the flag *m* is set, and will be simultaneously added to both **posT** and **negT** otherwise.
- Tweets are likely to contain words with the **opposite polarity**: we believe that unsetting the flag will produce instances with better **generalisation** properties.
- **Sample**: randomly sample with replacement *a* tweets from either **posT** or **negT** for each generated instance.
- **Averaging**: average and label sampled feature vectors.
- We create balanced training datasets with size equal to 1% of the size of the source corpus ($20,000$ in our experiments).

ASA results

	6HumanCoded		Sanders		SemEval	
EAA_U	0.805 ± 0.005	= = - -	0.800 ± 0.017	= = + +	0.802 ± 0.006	= + - -
EAA_B	0.809 ± 0.001	= = = =	0.795 ± 0.016	= = + +	0.798 ± 0.007	- = - -
LAA_U	0.809 ± 0.001	+ = = =	0.778 ± 0.002	- - = =	0.814 ± 0.000	+ + = =
LAA_B	0.809 ± 0.001	+ = = =	0.778 ± 0.003	- - = =	0.813 ± 0.001	+ + = =
ASA ($a = 1$, $m = T$)	0.806 ± 0.003	= = - -	0.786 ± 0.007	- - + +	0.808 ± 0.002	+ + - -
ASA ($a = 5$, $m = T$)	0.809 ± 0.002	= = = =	0.787 ± 0.005	- = + +	0.810 ± 0.003	+ + - -
ASA ($a = 10$, $m = T$)	0.804 ± 0.001	= - - -	0.776 ± 0.008	- - = =	0.806 ± 0.003	+ + - -
ASA ($a = 50$, $m = T$)	0.756 ± 0.003	- - - -	0.697 ± 0.005	- - - -	0.763 ± 0.002	- - - -
ASA ($a = 100$, $m = T$)	0.729 ± 0.002	- - - -	0.672 ± 0.006	- - - -	0.739 ± 0.002	- - - -
ASA ($a = 500$, $m = T$)	0.696 ± 0.003	- - - -	0.642 ± 0.008	- - - -	0.707 ± 0.005	- - - -
ASA ($a = 1000$, $m = T$)	0.690 ± 0.004	- - - -	0.637 ± 0.008	- - - -	0.701 ± 0.006	- - - -
ASA ($a = 1$, $m = F$)	0.793 ± 0.005	- - - -	0.762 ± 0.016	- - - -	0.787 ± 0.007	- - - -
ASA ($a = 5$, $m = F$)	0.837 ± 0.004	+ + + +	0.807 ± 0.010	= = + +	0.833 ± 0.003	+ + + +
ASA ($a = 10$, $m = F$)	$\mathbf{0.845} \pm 0.001$	+ + + +	$\mathbf{0.812} \pm 0.015$	+ + + +	$\mathbf{0.840} \pm 0.003$	+ + + +
ASA ($a = 50$, $m = F$)	0.815 ± 0.003	+ + + +	0.759 ± 0.006	- - - -	0.810 ± 0.004	+ + - -
ASA ($a = 100$, $m = F$)	0.781 ± 0.003	- - - -	0.720 ± 0.007	- - - -	0.779 ± 0.004	- - - -
ASA ($a = 500$, $m = F$)	0.723 ± 0.002	- - - -	0.670 ± 0.008	- - - -	0.729 ± 0.005	- - - -
ASA ($a = 1000$, $m = F$)	0.712 ± 0.002	- - - -	0.665 ± 0.007	- - - -	0.721 ± 0.005	- - - -

Table: AUC measure for different distant supervision models. Best results per column are given in bold.

ASA results

	6HumanCoded		Sanders		SemEval	
EAA_U	0.576 ± 0.007	= - - -	0.506 ± 0.018	= - - -	0.591 ± 0.018	= - - -
EAA_B	0.735 ± 0.008	+ = + +	0.709 ± 0.018	+ = = =	0.711 ± 0.006	+ = - =
LAA_U	0.729 ± 0.004	+ - = +	0.711 ± 0.003	+ = = +	0.725 ± 0.002	+ + = +
LAA_B	0.719 ± 0.002	+ - - =	0.703 ± 0.004	+ = - =	0.712 ± 0.002	+ = - =
ASA ($a = 1$, $m = T$)	0.734 ± 0.005	+ = + +	0.721 ± 0.010	+ + + +	0.724 ± 0.004	+ + = +
ASA ($a = 5$, $m = T$)	0.745 ± 0.005	+ + + +	0.723 ± 0.010	+ + + +	0.722 ± 0.006	+ + = +
ASA ($a = 10$, $m = T$)	0.737 ± 0.003	+ = + +	0.703 ± 0.011	+ = - =	0.708 ± 0.007	+ - - =
ASA ($a = 50$, $m = T$)	0.693 ± 0.003	+ - - -	0.643 ± 0.004	+ - - -	0.639 ± 0.006	+ - - -
ASA ($a = 100$, $m = T$)	0.672 ± 0.004	+ - - -	0.620 ± 0.005	+ - - -	0.607 ± 0.006	+ - - -
ASA ($a = 500$, $m = T$)	0.638 ± 0.004	+ - - -	0.599 ± 0.008	+ - - -	0.563 ± 0.005	- - - -
ASA ($a = 1000$, $m = T$)	0.635 ± 0.004	+ - - -	0.594 ± 0.010	+ - - -	0.554 ± 0.003	- - - -
ASA ($a = 1$, $m = F$)	0.717 ± 0.007	+ - - =	0.691 ± 0.013	+ - - -	0.699 ± 0.008	+ - - -
ASA ($a = 5$, $m = F$)	0.755 ± 0.004	+ + + +	0.730 ± 0.008	+ + + +	0.735 ± 0.005	+ + + +
ASA ($a = 10$, $m = F$)	$\mathbf{0.761} \pm 0.003$	+ + + +	$\mathbf{0.735} \pm 0.015$	+ + + +	$\mathbf{0.742} \pm 0.006$	+ + + +
ASA ($a = 50$, $m = F$)	0.749 ± 0.004	+ + + +	0.673 ± 0.005	+ - - -	0.699 ± 0.009	+ - - -
ASA ($a = 100$, $m = F$)	0.717 ± 0.003	+ - - -	0.645 ± 0.006	+ - - -	0.664 ± 0.005	+ - - -
ASA ($a = 500$, $m = F$)	0.665 ± 0.002	+ - - -	0.621 ± 0.007	+ - - -	0.621 ± 0.004	+ - - -
ASA ($a = 1000$, $m = F$)	0.653 ± 0.003	+ - - -	0.619 ± 0.007	+ - - -	0.613 ± 0.002	+ - - -

Table: Macro-averaged F1 measure for different distant supervision models. Best results per column are given in bold.

Conclusions

- The methods developed in this thesis can be used to **acquire** and **exploit** lexical knowledge for Twitter sentiment analysis under **label sparsity conditions**.
- We proposed two methods (Word Sentiment Associations and TCM) for building Twitter-specific **opinion lexicons** (acquisition of lexical knowledge).
- These methods could be used to create **domain-specific** lexicons.
- They could also be used to study the **dynamics** of opinion-words.
- Future work: try **non-linear representations** on TCM (Auto-Encoders or RBM).

Conclusions

- We proposed two **distant supervision methods** (TCM and ASA) that outperformed LAA and EAA for MPC.
- TCM is a **unified model** for message-level and word-level sentiment classification.
- Future work: subjectivity, emotions, handle negations, non-linear representations and deep networks. Volunteers?

Publications

1. F. Bravo-Marquez, E. Frank, and B. Pfahringer *Positive, Negative, or Neutral: Learning an Expanded Opinion Lexicon from Emoticon-annotated Tweets*, In *IJCAI '15: Proceedings of the 24th International Joint Conference on Artificial Intelligence*. Buenos Aires, Argentina 2015.
2. F. Bravo-Marquez, E. Frank, and B. Pfahringer *From Unlabelled Tweets to Twitter-specific Opinion Words*, In *SIGIR '15: Proceedings of the 38th International ACM SIGIR Conference on Research & Development in Information Retrieval*. Santiago, Chile 2015.
3. F. Bravo-Marquez, E. Frank, and B. Pfahringer *Building a Twitter Opinion Lexicon from Automatically-annotated Tweets*, In *Knowledge-Based Systems*. Volume 108, 15 September 2016, Pages 65 -– 78.
4. F. Bravo-Marquez, E. Frank, and B. Pfahringer *Annotate-Sample-Average (ASA): A New Distant Supervision Approach for Twitter Sentiment Analysis*, In *ECAI'16: Proceedings of the biennial European Conference on Artificial Intelligence*. The Hague, Netherlands 2016.
5. F. Bravo-Marquez, E. Frank, and B. Pfahringer *From opinion lexicons to sentiment classification of tweets and vice versa: a transfer learning approach*, In *WI'16: Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*. Omaha, Nebraska, USA 2016.
6. F. Bravo-Marquez, E. Frank, S. Mohammad, and B. Pfahringer *Determining Word–Emotion Associations from Tweets by Multi-Label Classification*, In *WI'16: Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*. Omaha, Nebraska, USA 2016.

Post PhD Projects

1. AffectiveTweets: and open source project for analyzing affect from social media posts (it has been used in around 30 publications and its website receives around 10 unique visits per day).
2. Emotion Intensity Detection.
3. Hate Speech Detection with deep neural networks.
4. Time-evolving word vectors.
5. Learning compatible representations between words and tweets.
6. Māori loanwords on Twitter.

Questions?

Thanks for your Attention!
#ThankYouHeaps
#GraciasTotales

Acknowledgements

- University of Waikato Doctoral Scholarship
- Machine Learning Group at the University of Waikato

../../img/waika

References I