Advanced Human Machine Interaction
Interaction Data Analysis

Textual Interaction Analysis

Alexandre Pauchet

alexandre.pauchet@insa-rouen.fr - BO.B.RC18

Normandie Université

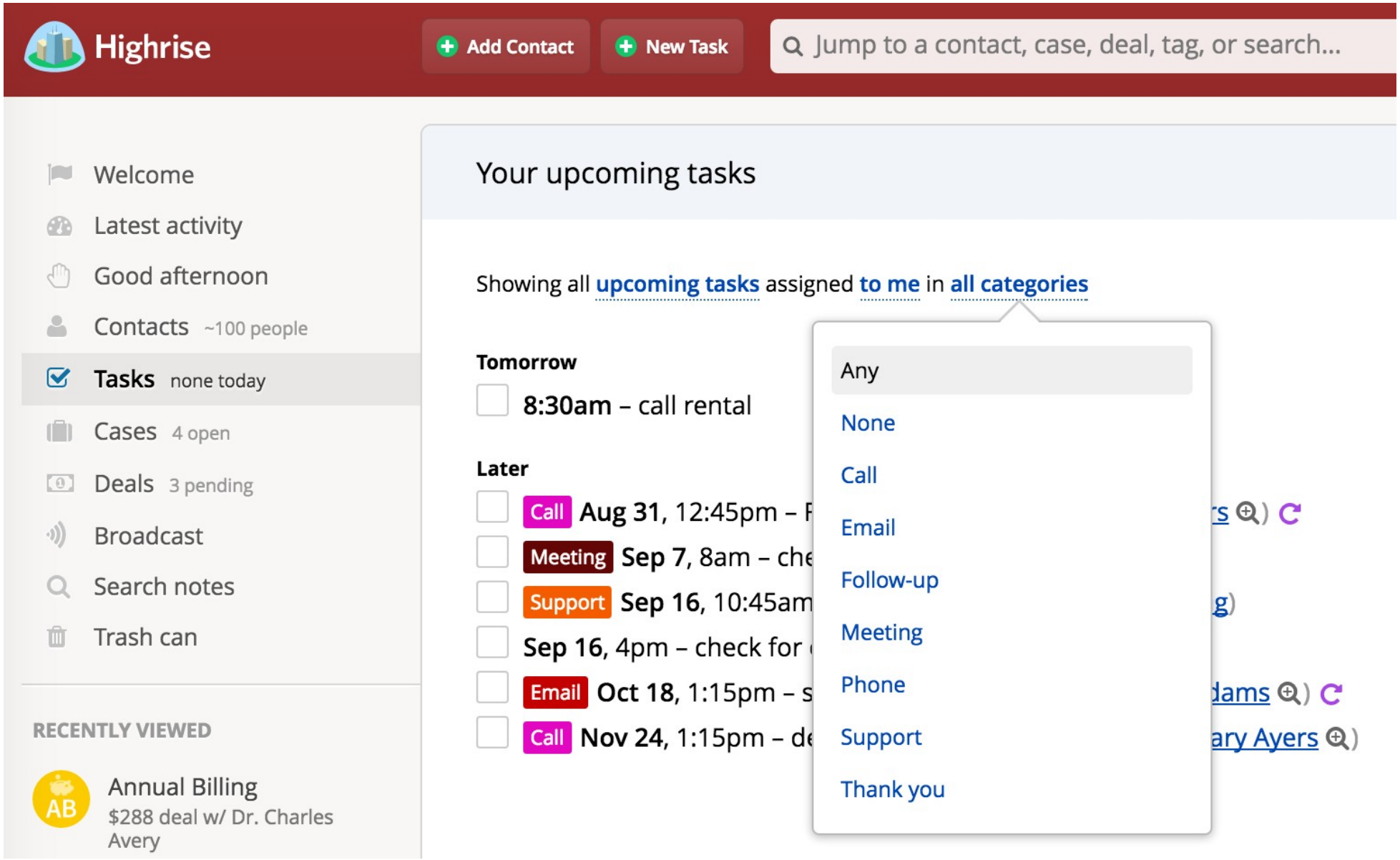
INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES ROUEN NORMANDIE

litis

Scope

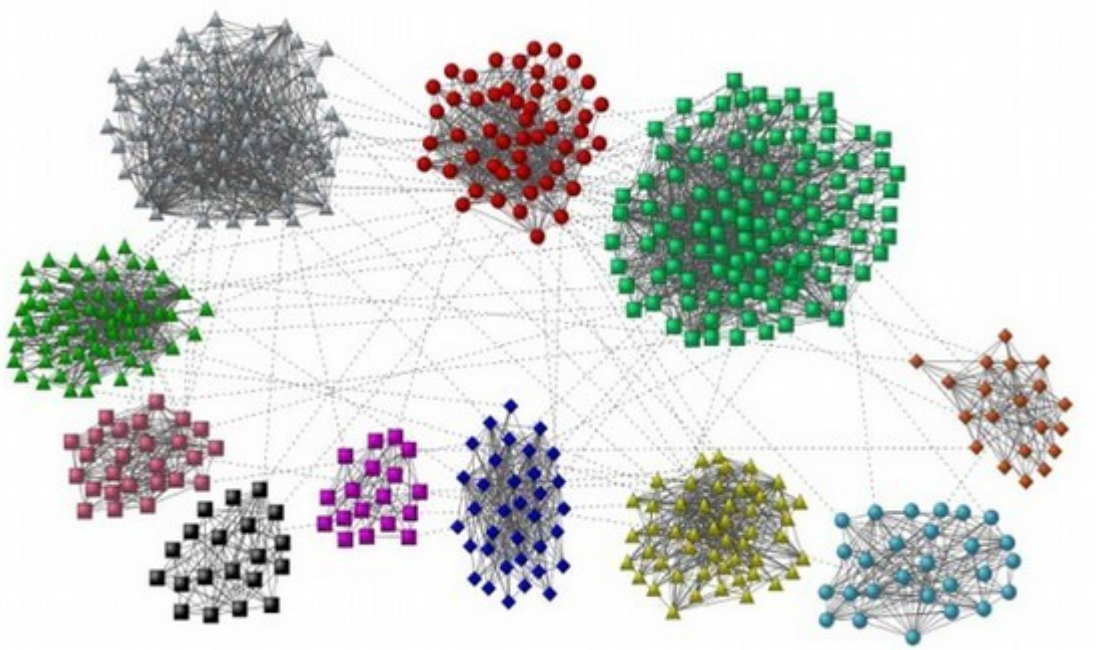
- **Type of textual interactions:**
  - Human-(mediated)-Human interactions: forums, mails, social network messages, blogs, chat, dialogues in online games, …
  - Human-machine interactions: (multi-modal) dialogue
- **Application goals:** filtering / labeling / sorting, opinion mining / sentiment analysis / affective computing, community detection, user assistance, companioning, serious games / virtual environments for learning, ...
- **Scientific problems:** supervised machine learning, unsupervised machine learning / data-mining, natural language processing (NLP) / natural language understanding (NLU) / natural language generation (NLG)

Example

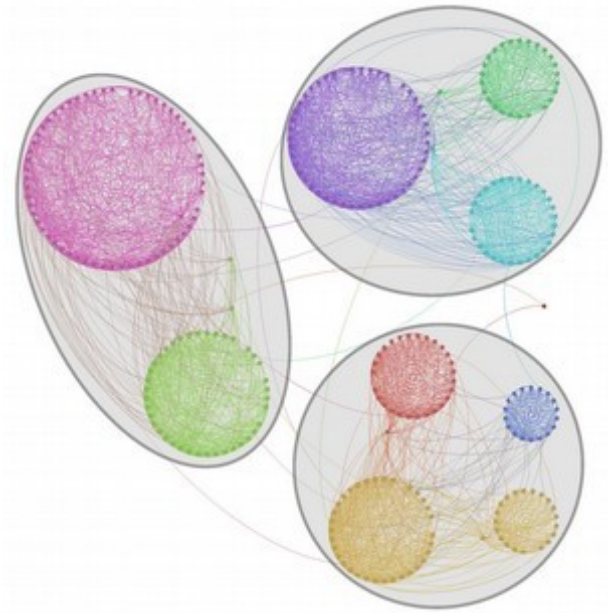


Automatic labeling of emails (supervised learning)

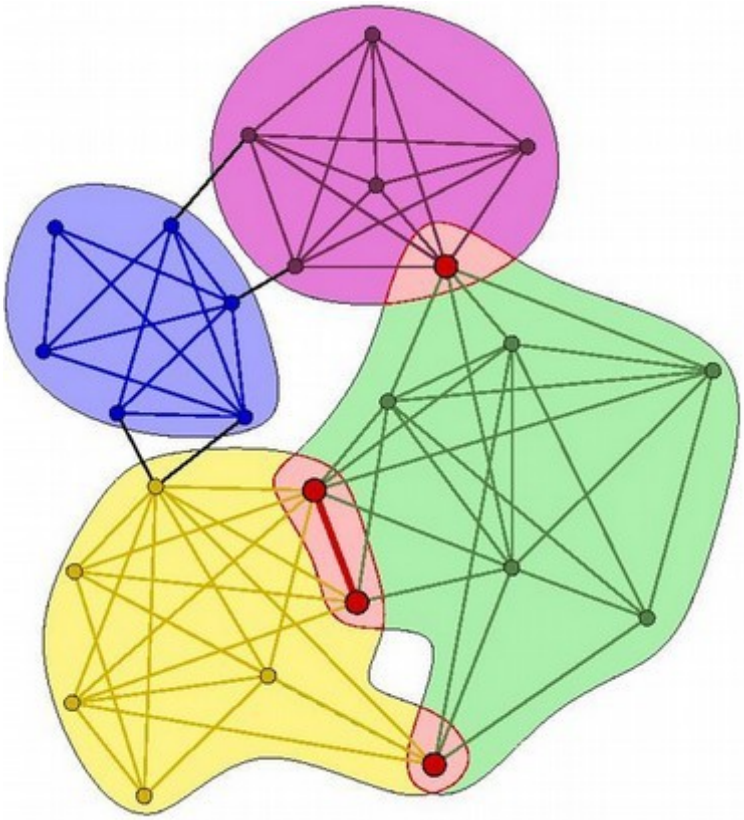
Example



Community detection in online social networks (unsupervised learning)

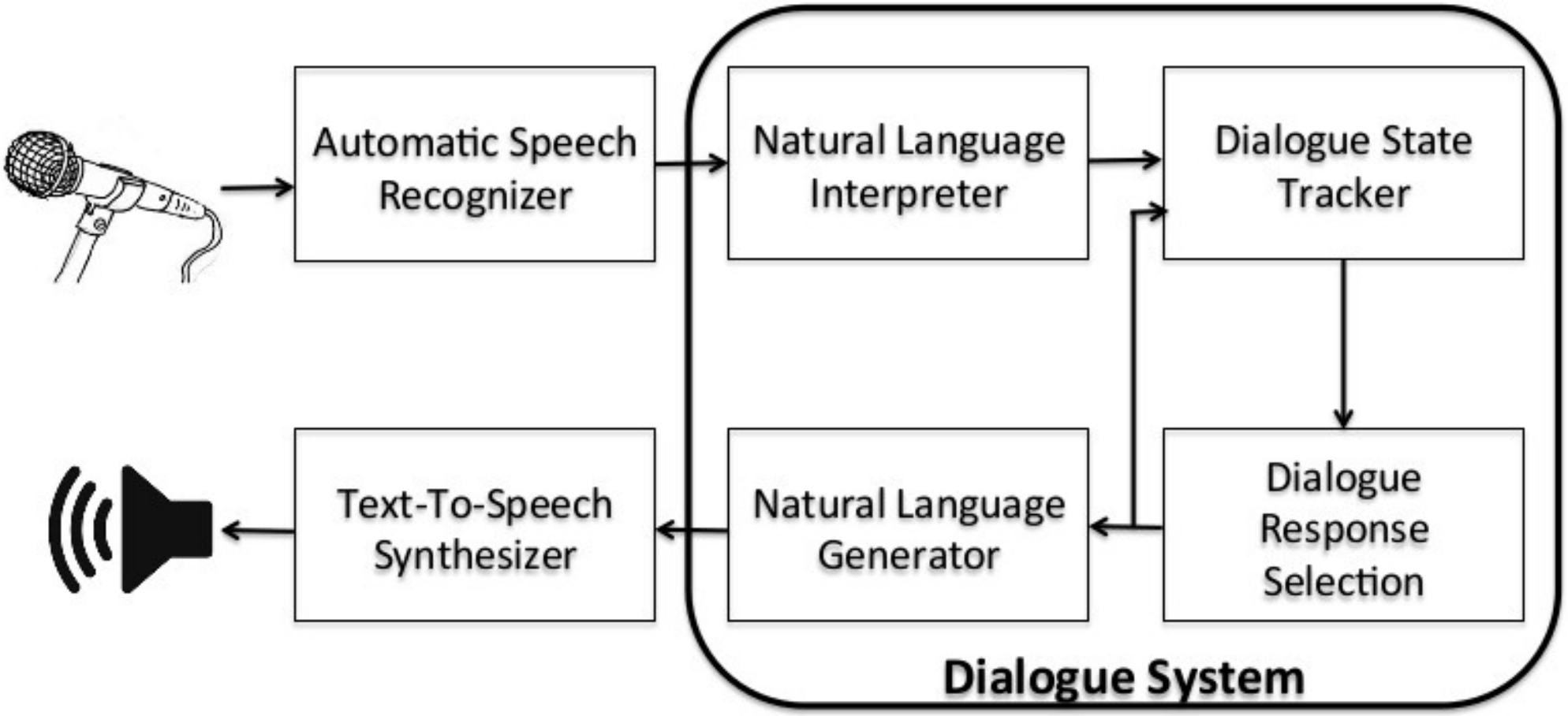






Example

Dialogical assistant (supervised learning, NLP, NLU, NLG)



I. V. Serban, R. Lowe, P. Henderson, L. Charlin, J. Pineau, "*A Survey of Available Corpora for Building Data-Driven Dialogue Systems*", https://arxiv.org/pdf/1512.05742.pdf

Exercise

Formalize the problem to automatically assign one or more predefined topics to a set of archived messages. Identify:

- the input;
- the expected output;
- the class of problem;
- the data required to learn the model.

Exercise

- Messages: Myear = $\{m_1, ..., m_n\}$
- Possible topics: T = $\{t_1, ..., t_m\}$
- Expected output:

  for each message m_i, predict a subset $T_i \subseteq T$

  $$f : \text{Myear} \rightarrow P(T)$$

  $$m_i \mapsto T_i$$

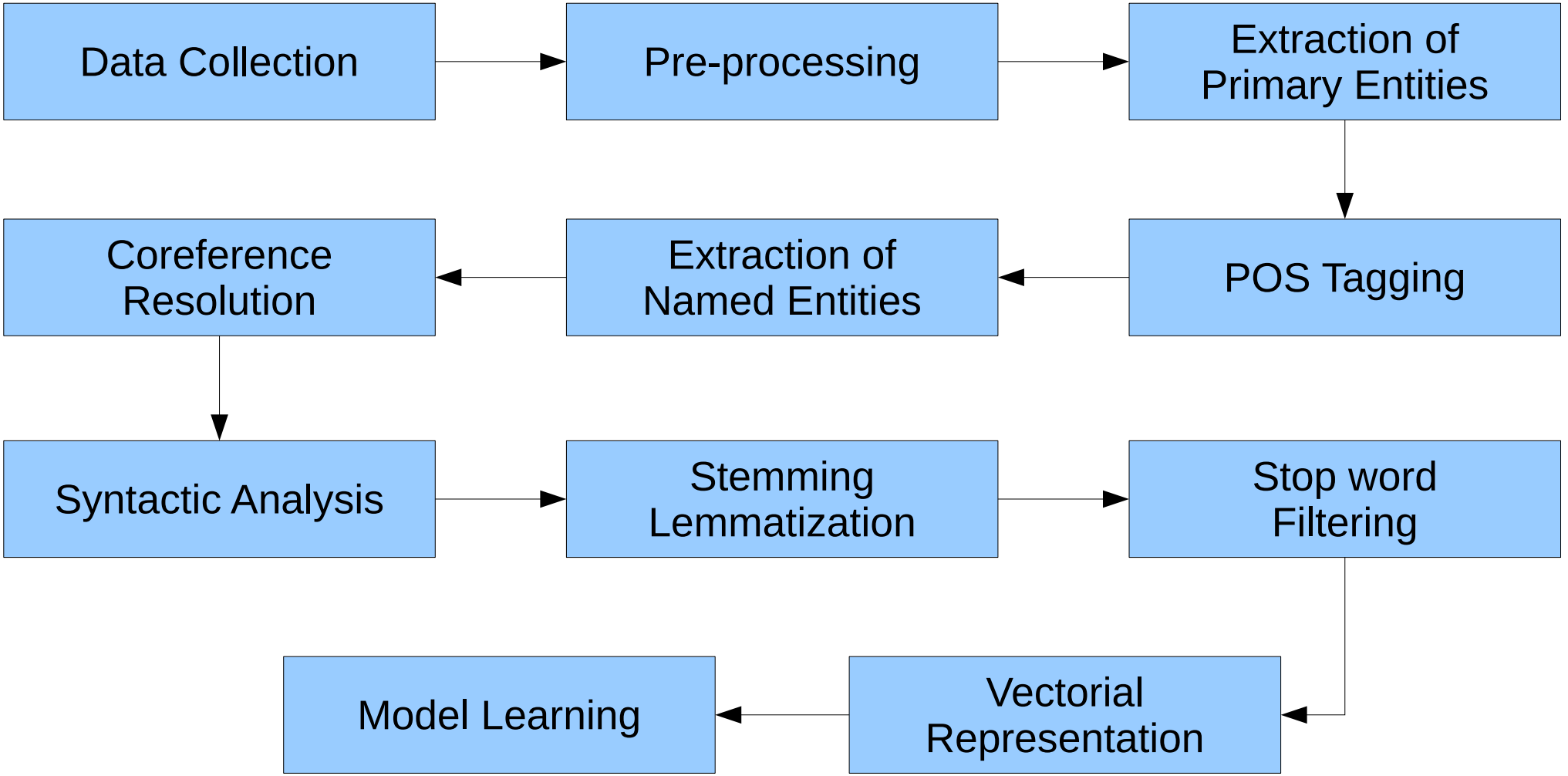
- Problem class: Supervised multi-label text classification
- Required training data:

  $D = \{(m_1, Y_1), ..., (m_x, Y_x)\}$, with $Y_i \subseteq T$

If the topics were not predefined, the problem would instead involve clustering or topic modelling.

Classification of (short) texts

Process (1/6)



The required steps depend on the task, the data and the chosen representation.

Process (2/6)

- **Textual data collection**:
  - Identify the sources and the unit of analysis (message, utterance, conversation, session, user, ...).
  - Preserve useful contextual information and metadata
  - Consider sampling, consent, privacy and licensing
- **Text processing**:
  - Encoding and Unicode normalisation
  - Treatment of URLs, mentions, emojis and special characters
  - Possible normalisation of spelling, abbreviations and informal language.
- **Tokenisation and linguistic units**: Split the text into sentences, tokens or subword units. The segmentation depends on the language, the task and the representation.
- **POS tagging** (part-of-speech): grammatical characterization of text components with a lexical category and a function. Current NLP pipelines include spaCy and Stanza.

(Stanza: https://stanfordnlp.github.io/stanza/pipeline.html)

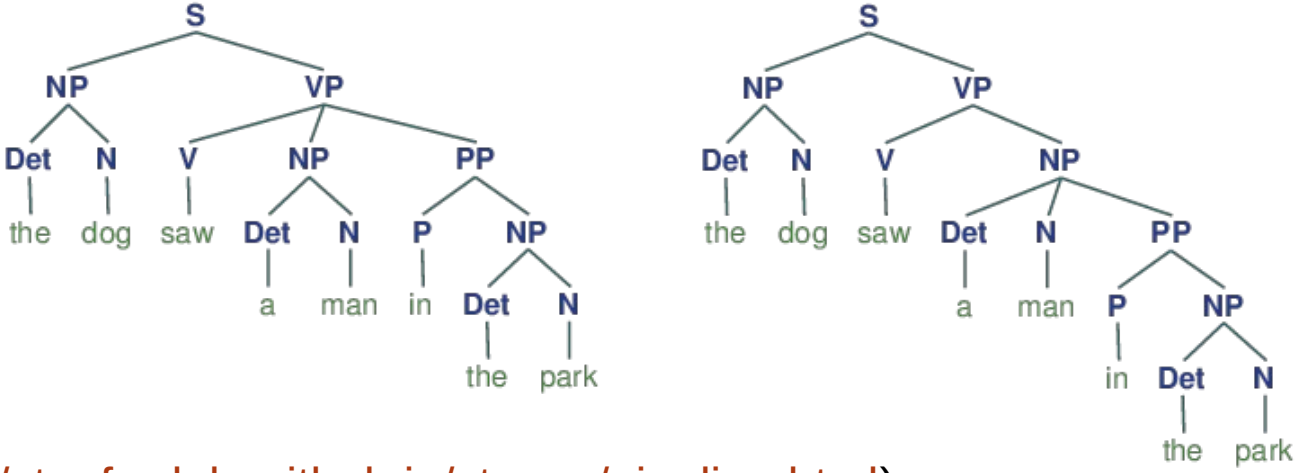
Process (3/6)

- **Named entity recognition**: Locate and classify mentions of persons, organisations, locations, dates, products, ...

  Difficulties: metonymy *(*ex: *Little Red Riding Hood, The White House will be making an announcement),* polysemy

- **Coreference resolution**: Identify expressions that refer to the same entity (he / she / it, his / her, …)

- **Syntactic analysis**: Identify grammatical relations between words. It can help represent phenomena such as the scope of negation



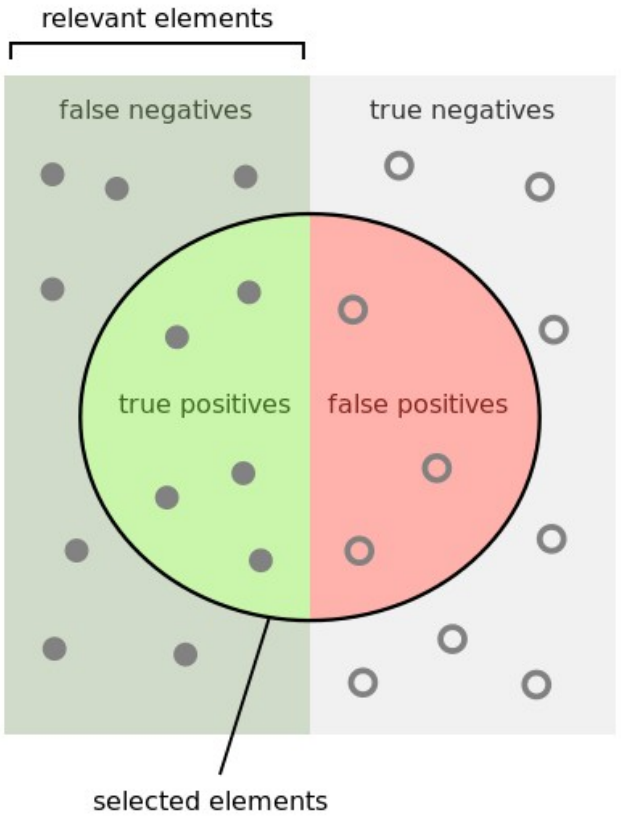
(Stanza: https://stanfordnlp.github.io/stanza/pipeline.html)

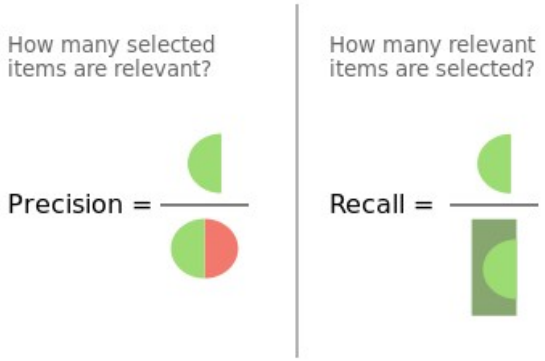
These analyses are useful for some tasks, but not required for every text classifier.

Process (4/6)

- **Stemming or Lemmatization:**
  - *Stemming*: reduce related words to a common stem using heuristic rules. The resulting stem is not necessarily a valid word.
  - *Lemmatization*: Map an inflected word to its dictionary form using linguistic information.
- **Stop words filtering**: Remove very frequent words that are unlikely to be useful for the task. This operation is task-dependent: removing negations such as “not” may damage sentiment classification.
- **Vectorial representation:**
  - *Principle*: each text (document, request, utterance) is represented by a large and sparse vector; bag of words approach.
  - *Possible dictionaries*: set of words of the corpora, smaller or larger external set of words, n-grams, …
  - Comparison of vectors using the cosine similarity
  - *Improved representations*: co-occurrence matrix, tf*idf, LSA, word embeddings (Word2Vec), language model (BERT-like model), ...
- **Learn the models**

Process (5/6)





- **Model evaluation**:
  - Precision

$$\text{precision} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{retrieved documents}\}|}$$

  - Recall

$$\text{recall} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{relevant documents}\}|}$$

  - F-measure $F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$

- **Remarks**:
  - Unbalanced classes!
  - Evaluation by class

https://en.wikipedia.org/wiki/Precision_and_recall

Process (6/6)

- **Evaluation protocol**
  - Separate training and test data before fitting the model or the text representation;
  - use a stratified split or stratified cross-validation;
  - examine the confusion matrix;
  - report results for each class and macro-F1 when classes are unbalanced.
  - The vocabulary and TF-IDF weights must be learned from the training data only.

TF*IDF

- **Objective**: weight the terms according to their "importance" in the selection process; particularly useful for information retrieval:
  - *Term Frequency*: the importance of a term for a document is proportional to the number of occurrences of the term in the document,
  - *Inverse document frequency*: the importance of a term for all documents is inversely proportional to the number of documents in which it appears (terms appearing in few documents are more discriminating that terms in many documents)
- **Definition**:

$$\mathrm{tfidf}(t,d) = \mathrm{tf}(t,d) \times \log\left(\frac{N}{\mathrm{df}(t)}\right)$$

  - tf(t,d): frequency of term t in document d
  - df(t): number of documents containing term t
  - N: total number of documents

Exercise

1) *Human-machine interfaces for computer applications*
2) *User opinion of computer system response time*
3) *User interface management system*
4) *System engineering to improve the computer response time*
5) *Complex systems made of humans, computers and agents*
6) *A dialogical agent is interacting with a human user on computer*

1. Define the preprocessing decisions:

- How should “human-machine” be tokenised?
- Should “human” and “humans” represent the same term?
- Which words should be removed?

2. After lowercasing and lemmatisation, construct the binary document-term matrix restricted to:

V = {human, computer, user, system, agent, response, time}

3. Compute df and idf for “human” and “computer”.

4. Which term is the most discriminative? Why?

Latent Semantic Analysis (LSA)

- **Objective**: Represent terms and documents in a lower-dimensional latent space based on their patterns of co-occurrence.
  - Removal of "noise" from data;
  - Replacement of lemmas by corresponding concepts.
- **Approach**:
  - Start from a document-term matrix M, often weighted with TF-IDF;
  - apply singular value decomposition: $M = U \cdot S \cdot V^T$
  - retain only the k largest singular values: $M \approx U_k \cdot S_k \cdot V_k^T$
- **Effects**:
  - reduce dimensionality and noise;
  - capture correlations between terms;
  - partially account for synonymy

Limitations: linear representation, no word order and no contextual meaning.

Word embeddings (Word2Vec)

- **Objective**:
  Learn a dense vector representation of each word from the contexts in which it occurs.

- **Two training objectives:**
  - CBOW predicts a word from its context;
  - Skip-gram predicts surrounding words from a given word.

- **Properties:**
  - words used in similar contexts receive similar vectors;
  - semantic relations may appear as vector regularities;
  - each word has a single representation, independently of context.

- **Remarks**:
  - With this representation, the words are grouped by similarity context
  - A form of "additivity" is possible:

$$\text{vec(King)} - \text{vec(Man)} + \text{vec(Woman)} = \text{vec(Queen)}$$

- **Ref**: T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, and J. Dean, "*Distributed representations of words and phrases and their compositionality*", In Advances in neural information processing systems, pp. 3111–3119, 2013.

Contextual representations (Transformers)

- **Static word embeddings:**
  One vector is associated with each word.

  bank → one vector

- **Contextual representations:**
  The representation depends on the complete sentence.

  “The bank approved the loan.”

  “We sat on the river bank.”

  bank → different contextual vectors

- Transformer language models are pre-trained on large text corpora.

- **They can be used through:**
  - contextual embeddings followed by a classifier;
  - fine-tuning for a specific task;
  - prompting for large generative models.

Pre-training improves the representation,
but task definition, training data and evaluation remain essential.

Sentiment, stance and emotion analysis

Definitions and objectives

- **Sentiment Analysis**:
  Identify an evaluation or polarity expressed toward a target.

  Examples:

  - positive / negative / neutral
  - or a continuous valence score (Valence [-1:1])

- **Emotion recognition**:
  Identify emotions expressed or described in a text.

  Examples:

  - Emotion={Anger, Disgust, Fear, Joy, Sadness, Surprise, Neutral}
  - or dimensions such as valence and arousal

- **Stance detection:**
  Determine whether the author is in favour of, against or neutral toward a specific target.

- **Opinion** = {holder, target, aspect, polarity, time};

B. Pang and L. Lee, "*Opinion mining and sentiment analysis*", Foundations and Trends in Information Retrieval, Vol. 2:1-2, pp. 1–135, 2008.

Process

- **Classic process**: textual data collection, pre-processing, POS tagging, stemming (en) / lemmatization (fr), stop words filtering, vectorial representation (bag of words + TF*IDF + LSA), supervised classification (SVM, Random Forests, NN)
- **Possible task formulations**:
  - Binary classification: positive / negative
  - Multiclass classification: positive / negative / neutral
  - Multilabel classification: several emotions may be expressed simultaneously
  - Regression: continuous valence or arousal
  - Target-dependent analysis: sentiment, aspect or stance toward a specified target
- **Possible methods**
  - Lexicon-based rules;
  - sparse representations with supervised classifiers;
  - pretrained language models.

Annotation may be subjective.
Measure and report inter-annotator agreement.

Stance versus Opinion

"***Sharknado 3 may be the best film I've seen yet. #Sharknado3 #America***"

- Target: Sharknado 3 ; polarity: POSITIVE
- Target: Sharknado 3 ; SUBJECTIVE → POSITIVE

"***@HilaryClinton going to prison. She can help #build-thewall.***"

- Target: Hilary Clinton ; polarity: NEGATIVE
- Stance about Donald Trump: POSITIVE

The expressed sentiment does not directly determine the stance.
Both depend on the target being analysed.

Activity: what should the model predict?

1)“Great, the application has crashed again.”

2)“I am worried that the robot will make another mistake.”

3)“The interface is beautiful, but far too slow.”

- For each message:
  - What is the target?
  - What sentiment or emotion is expressed?
  - What makes the annotation difficult?

Why affect analysis is difficult!

1) “Great, the application has crashed again.”
   Negative sentiment expressed through sarcasm.

2) “I am worried that the robot will make another mistake.”
   Fear or anxiety, directed toward a possible future event.

3) “The interface is beautiful, but far too slow.”

   Positive sentiment about appearance;

   Negative sentiment about performance.

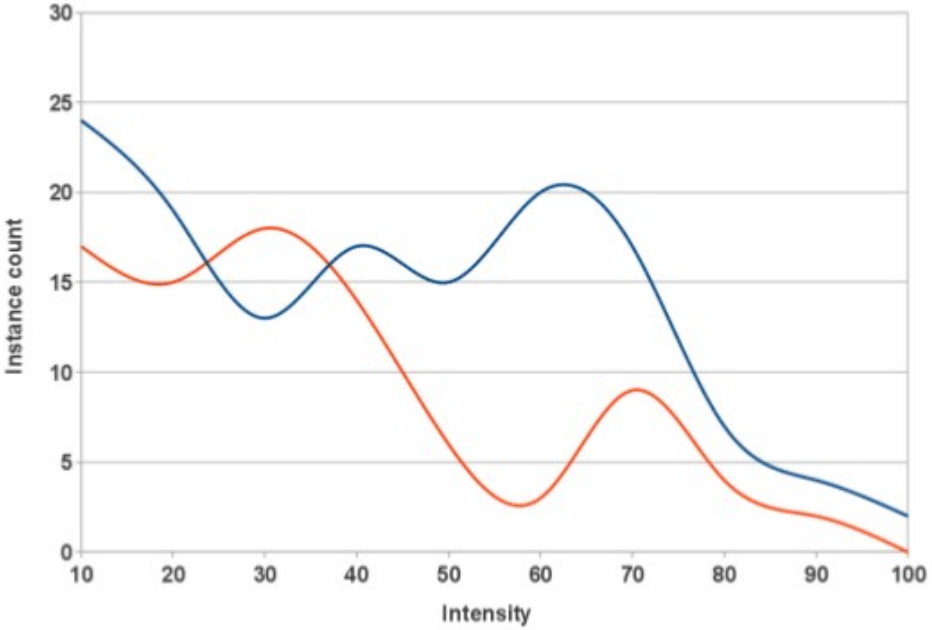
Main difficulties:

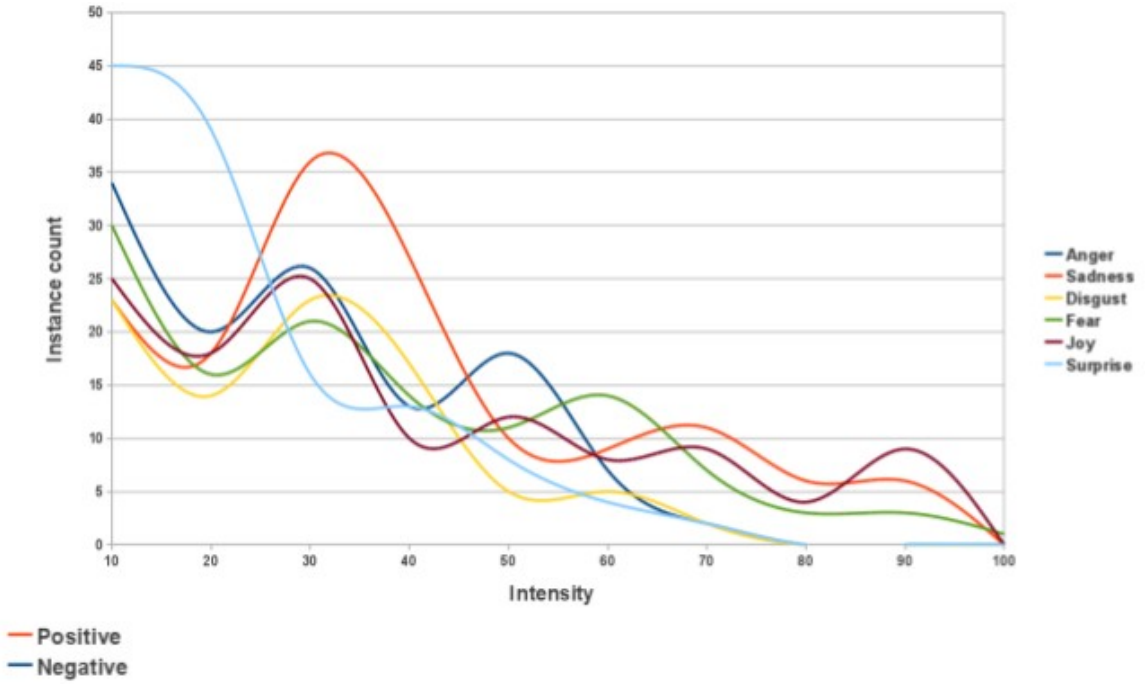
- the target and aspect must be identified;
- polarity may be implicit or sarcastic;
- several sentiments or emotions may coexist;
- annotations may legitimately differ.

Example: emotion detection

A	D	F	J	Sad.	Sur.	Headline
-	-	-	0.15	0.25	-	Bad reasons to be good
-	-	-	-	-	0.36	Martian Life Could Have Evaded Detection by Viking Landers
-	-	0.68	-	-	-	Hurricane Paul nears Category 3 status
-	-	-	0.75	-	0.57	Three found alive from missing Russian ship - report
0.52	0.64	0.50	-	0.43	-	Police warn of child exploitation online

Anger=**A**, Disgust=**D**, Fear=**F**, Joy=**J**, Sadness=**Sad.**, Surprise=**Sur.**



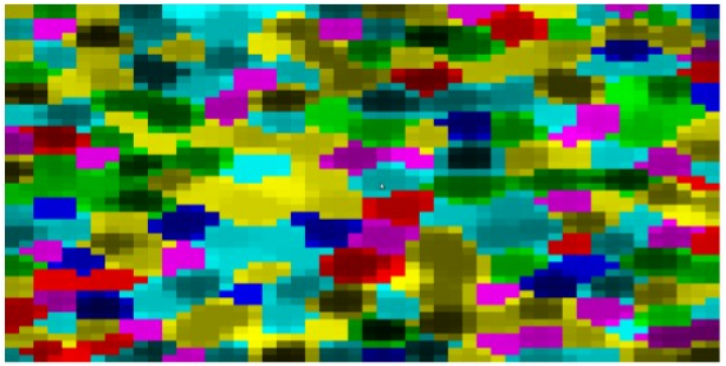


SemEval 2007, task 14 : Strapparava, C. and Mihalcea, R. (2008). Learning to identify emotions in text, In Proceedings of the 2008 ACM symposium on Applied computing, pp. 1556–1560, 2008.

Results

	LSA All emotional			UA			UPAR7		
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
Anger	6.20	**88.33**	11.59	12.74	21.60	16.03	16.67	1.66	3.02
Disgust	1.98	**94.12**	3.88	0.00	0.00	-	0.00	0.00	-
Fear	12.55	**86.44**	21.92	16.23	26.27	20.06	**33.33**	2.54	4.72
Joy	18.60	**90.00**	30.83	40.00	2.22	4.21	**54.54**	6.66	11.87
Sadness	11.69	**87.16**	20.62	25.00	0.91	1.76	**48.97**	22.02	**30.38**
Surprise	7.62	**95.31**	14.11	13.70	16.56	**14.99**	12.12	1.25	2.27

Classifier: self organizing map (SOM)



Nb. of instances		
No emotion	642	64.85%
Anger	14	1.41%
Disgust	6	0.61%
Fear	65	6.57%
Joy	110	11.11%
Sadness	81	8.18%
Surprise	38	3.84%
Combined	34	3.43%

	Precision	Recall	F1
LSA training	20.50	19.57	20.02
LSA Gutenberg	24.22	23.31	**23.76**
LSA All emotion	9.77	**90.22**	17.63
UA	17.94	11.26	13.84
UPAR7	**27.60**	5.68	9.42

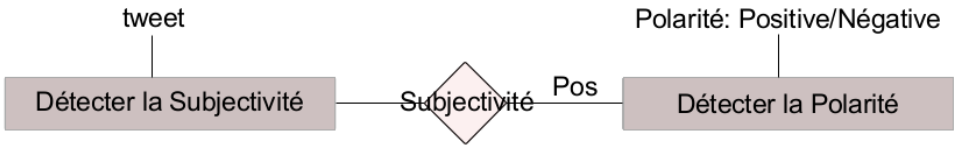
	LSA training			LSA Gutenberg		
	Prec.	Rec.	F1	Prec.	Rec.	F1
Anger	10.00	11.86	10.85	**18.52**	15.38	**16.80**
Disgust	3.33	4.17	3.70	**8.33**	7.69	**8.00**
Fear	19.01	17.76	18.36	28.39	27.67	**28.03**
Joy	36.75	36.75	36.75	40.49	64.62	**49.79**
Sadness	24.14	40.00	30.11	27.08	19.60	22.74
Surprise	**29.73**	6.92	11.23	22.50	4.95	8.11

Example: Opinion mining on tweets

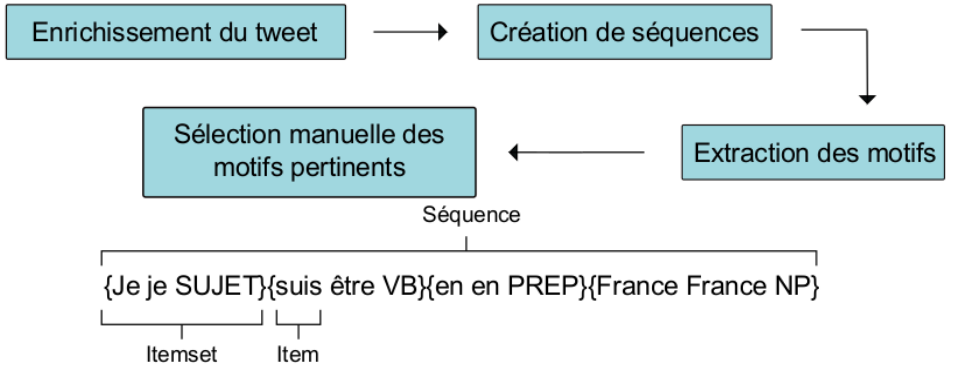
Corpus de tweets

Corpus d'apprentissage :
Chaque tweet est annoté automatiquement suivant les émoticônes qu'il contient : polarité positive (:-p,:-D...) ou négative (:-(, :-s...).
Résultat : 300 000 tweets français et 300 000 tweets anglais.
Corpus de test :
Annotation manuelle par des ingénieurs.
Résultat : 800 tweets anglais et 700 tweets français.

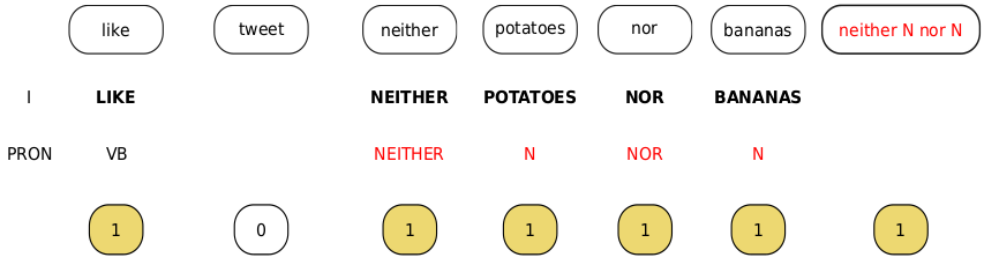
Méthode proposée



Détection de subjectivité



Détection de polarité



Résultats obtenus

Détection de subjectivité (motifs séquentiels fréquents) :

	précision	rappel
français	65%	66%
anglais	64%	62%

Détection de polarité (SVM) :

		précision	rappel
PRESENCE	français	62%	61%
	anglais	74%	73%
POSITION	français	62.6%	57%
	anglais	60%	60%

Textual interaction analysis: key decisions

1) Define the interaction phenomenon and the unit of analysis:
   message, utterance, conversation, user or session.

2) Define the expected output:
   class, multiple labels, score, extracted entities or generated text.

3) Choose a representation suited to the task:
   sparse vectors, static embeddings or contextual representations.

4) Select a method and an evaluation protocol consistent with the
   available annotations and the intended use.

A more complex language model cannot compensate for an ill-defined task or an invalid evaluation protocol.