MSDS 7337 - NLP - Glossary

Patrick McDevitt

Masters of Science in Data Science, Southern Methodist University,
Dallas, TX 75275 USA
pmcdevitt@smu.edu

Abstract. This glossary is a catalog of interesting terms encountered in the study of the SMU NLP course during fall semester 2018.

> It is more fun to talk with someone
> who doesn't use long, difficult
> words but rather short, easy
> words, like 'What about lunch?'
>
> *A. A. Milne*

Note : the information in this glossary relies heavily upon the videos and support presentation materials provided by Prof Timothy Musgrove, Ph.D. in the Fall 2018 MSDS 7337 NLP class. As the intent of this document is to fulfill a class requirement (and no external distribution is planned) there are no specific attributions to the class materials identified within. Where external sources were used, references are identified.

1 Unit 01

- Goal for attributes :
  - jointly sufficient - not too broad
  - individually necessary - not too narrow
- NLP - natural language processing - NLU + NLG
- NLU - natural language understanding - We try to get the machine to produce a useful representation of some inputted natural language

- NLG - natural language generation - We try to get the machine to produce usable, natural language output that is not just identical to its input.
- Applications of NLU
  - Text annotation - tagging, meta-data extraction/generation, classification, document summarization
  - Corpus analytics - theme extraction, clustering, taxonomy mapping, sentiment analysis
  - Search applications - query repair, query refinement, results post processing (ranking, clustering, encapsulation)
  - Advanced applications - machine translation, knowledge discovery, question handling
- Applications of NLG
  - Text annotation - document summarization, generation of call outs / headlines
  - Corpus analytics - labelling of clusters, synopsizing of corpus-wide topic and/or sentiment trends
  - Search applications - advanced capsule generation (summarization modified to fit the query), advanced query refinement (next gen version for disambiguation)
  - Advanced applications - machine translation, knowledge discovery, question handling (refinement, answering)

2 Unit 02 - Levels of Analysis in NLP

- Levels of NLP analysis
  - Lexical Analysis – Words
    - lexicon - dictionary, vocabulary
    - morphology - study of morphemes - units of which words are made
    - stemmer - algorithm that provides root morpheme
    - metadata - corpus derived meta-data supports developing : word frequency score, common collocation, and commonly co-occurring words and context words
    - collocations - words commonly occurring together
    - head word - specific listing in the lexicon
    - polysemous - more than one meaning

  * WordNet - the Bible of word senses, produced by Cognitive Science Laboratory of Princeton University - Psychology Dept -, shows relationships between words. many real world senses are not in WordNet (i.e, apocryphal)
  * domain relations - mode of discourse around certain activity or subject matter
  * Uses of lexical analysis - text and corpus analytics : spell correction, terminology extraction (OCR), lexical diversity measurement

- Syntactic analysis – Grammar

  * sentence boundary detection - needed since syntax analysis is sentence by sentence analysis. grammar parsing relies on sentence boundary detection. non-trivial problem.
  * part of speed (POS) - each word in a sentence can be assigned to a part of speech. most words have more than one part of speech; therefore, POS tagging is non-trivial. needs contextual clues - words preceding and succeeding,
  * Penn Treebank Tagset - commonly used tag sets for parts of speech, approx 40 different POS with this set, support eventual grammar parsing
  * parsing - break the sentence into grammar parts
  * Lemmatizaton - canonical (conventional) form that represents a set of related word forms
  * discrete text field analysis - unitizing, normalizing, smart ETL.

- Semantic analysis – Meanings

  * named-entity extraction - NEE or NER - recognize entity without typing - persons, organizations, places, events. good NEE will cluster together many variants, including epithets. not as simple as looking up entities in wikipedia
  * relationship extraction - need syntax analysis and semantics - need a representation of the world (an ontology)
  * - word sense disambiguation - still unsolved problem in AI - how to distinguish the meaning of a word in context, from all of the possible meanings of the word
  * classification - use a tree-structured graph to place documents into categories - often use machine learning (such as SVM)

    - taxonomy - hierarchical structure in which each entity is assigned to one category. IAB is likely most influential taxonomy - Internet Advertising Bureau - is huge part of internet experience.
  - Discourse / Entailment Analysis – Inferences

3 Unit 03 - Trade-offs in various approaches - NLP

- shallow vs. deep - deep semantically (every nuance of every word and phrase) or shallow treatment (scrape surface of documents for meta-data and high level summaries)
  - historically, in machine learning, deep refers to network with several hidden layers
  - more commonly, deep refers to deep or shallow parsing - shallow parsing - for example - leaves sentence in chunks (such as noun phrases) while deep parsing breaks down every phrase/sub-phrase/word to part of speech
  - Stanford has a great parser - focuses on verb as center of action and then identifies who is actor and object of the verb - but is still shallow semantic parsing since not every word is included in the analysis
  - - tagging, topic segmentation, semantic analysis – all shallow semantics
  - - shallow - more feasible, less computational time, less complexity
  - - deep & shallow : both are currently significantly used
- statistical vs. symbolic - complex statistical methods or rules of logic ... statistics is current popular but logic may return
  - symbolic - symbols (logic) rule based system to apply criteria to textual input. build entire system to classify documents. rules are explainable to whomever understands logic — seems similar to human reasoning – this is manually controllable - which means harder to maintain - suffers from the challenge of human complex thinking (deeply) about meaning
  - statistical - compute statistics – more easily scalable - just implements based on stats - does not question its own logic
- feature engineering vs. feature learning - human experts to engineer features or have machine learn features ... feature learning is dominant trend (currently)
  - best of both worlds ...
    - bootstrap candidate features

  - SMEs validate and annotate features
  - grade the classification results and repeat same cycle until performance achieved

- top-down vs. bottom-up - start with high level of classifications and gradually break them down into details .. or do the inverse
- transparent vs. opaque - AI vs XAI (explainable AI) - transparent means it is easy enough to see what the algorithm is doing; opaque is result of more complex machine learning and produces results not easily explainable

4 Unit 04 : Working in NLP

- declared data & inferred data - declared is what is directly stated in response to a question or situation;inferred data is an alternative view using non-direct methods; inferred data is useful to verify the declared data. e.g., direct survey responses to questions (declared) vs. characterizing the same respondent's information available from social media sites to identify if they are correlated or contraindicating

5 Unit 05 : Low-Level Analysis

- Primary features - extracted from the primary document : word frequencies and collocations (n-grams)
- Secondary features - comparison features between / among documents
  - Differential frequency analysis
    - most common technique is tf-idf
    - compare frequency of term / phrase in one document vs. a collection of documents - provides relative measure of whether a term is more 'important' in one document than in companion documents
    - tf-idf = term frequency - inverse document frequency
  - Relative lexical diversity
    - often measured via type-token ratio : (number of unique tokens) / (number of tokens)
    - essentially, count diversity of vocabulary in comparison to other documents

  - based on homework 1 - not a very robust metric as stand-alone measure of diversity, but can be improved by accounting for additional elements such as document length and also by incorporating a sufficiently large set of documents in the corpus as a comparative set

- Reading level
  - sentence length, average number of syllables, words outside of a baseline list

- Normalization
  - stop words - common words that are repeated in all or most text and therefore are not useful for distinguishing differences between texts, e.g., a, an, the, on, of, etc. these words are a necessary ingredient in text normalization
  - content vs. function words - function words are considered a fixed, defined set of words, and they do not contribute to specific understanding of the text content by providing responses to questions like : who, what, when, where, why. content words, on the other hand, are an open class (continually evolving set) and do provide responses to the W questions. content words typically far outnumber the function words, and the two classes of words are mutually exclusive. function words comprise the base list of stop words.
  - misspellings - managed in two ways - edit-distance method and fuzzy string comparison. the edit distance method compares a detected mis-spelled word to known potential corrections and determines the distance to candidate words by counting the number of edits (additions, deletions) needed to transform the mis-spelled word to each of the candidate corrections the choice with the smallest edit distance is chosen as the preferred replacement. the fuzzy string approach completes a comparison of misspelled word and candidate replacements while maintaining character order, and compares number of characters in sequence that match, and prefers the replacement with longest substring length.
  - stemming - with the aid of a morphology (smallest elements of words, the grammatical atoms), a stemmer reduces each word to its smallest identifiable unit. in this way variants of a word are all reduced to a common stem

6 Unit 06 : Lexical Knowledge Bases

- hyponym / hypernym -
  - a hypernym is a word whose meaning includes the meanings of other words. flower is a hypernym of daisy and rose. hypernyms (also called superordinates and supertypes) are general words; hyponyms (also called subordinates) are subdivisions of more general words. The semantic relationship between each of the more specific words (e.g., daisy and rose) and the more general term (flower) is called hyponymy or inclusion. (https://www.thoughtco.com/hypernym-words-term-1690943)
  - a meronym is a word that denotes a constituent part or a member of something. apple is a meronym of apple tree. The opposite of a meronym is a holonymthe name of the whole of which the meronym is a part. – Apple tree is a holonym of apple (https://www.thoughtco.com/what-is-a-meronym-1691308)
- What is WordNet ??
  - lexical databases derived from the original Princeton WordNet
  - groups words into synonym sets and interlink them using lexical and conceptual-semantic relations
  - a combination of dictionary and thesaurus - intuitively usable - supports automatic text analysis and artificial intelligence applications
  - WordNet was developed by the Cognitive Science Laboratory at Princeton University under the direction of Professor George A. Miller (Principal Investigator)
  - WordNet is considered to be the most important resource available to researchers in computational linguistics, text analysis
  - (https://www.igi-global.com/dictionary/ontology-based–language-learning/32725)

7 Unit 07 : Syntactic Analysis: POS-Tagging

- part-of-speech (POS) tagging - quite simply : identifying the part of speech (noun, verb, adjective, etc.) associated to each word in a text
- implementation : WordNet is an elementary resource to implement POS tagging, but wordNet has only four parts of speech, so it is a bit too fundamental for most useful purposes. There are several POS taggers of different complexity. The most widely used (currently) is Penn Treebank, which allows for forty-five tags. In addition to the basic four of WordNet, the Penn

Treebank set includes some additional distinctions to further refine noun terms (four noun distinctions), verb forms (six verb distinctions), and similarly for adjectives, adverbs, and also characterized functional words such as prepositions and articles.

- how well does it work ? - the Penn Treebank provides approx 93% accuracy, by some estimates. the best taggers in use achieve approx 96% accuracy, and the best humans do not agree more definitively than that. so, automated tagging is at approximately the same level of accuracy as can be achieved even with expert human tagging.

8 Unit 08 : Syntactic Analysis: Parsing

- parsing can be considered at different levels : shallowest level is part-of-speech tagging, next level is to parse to small phrases that represent groups of parts of speech, such as noun phrase or verb phrase, and at the most expansive level is full grammar tree. depending upon the intended purpose of the parsing, an appropriate level of parsing can be chosen.
- full grammar parsing can be complex, as sentence structure (at least in English) provides possibility for highly cmplex structures.
- chunking - or shallow parsing - has advantages over full parsing. these advantages include :
  - chunking is less computationally expensive
  - full parsing may not be accurate - there can be challenges for extraordinalriy complex sentence strutuctures, or moore typically, user generated content, content generated by non-native speakers, and modern communications styles of short text messages all contribute to potential to have inaccuracies in full parsing anyway
  - often – full parsing is not needed for intended purpose. full parsing is beneficial in the context of full translation from one language to another - where full meaning is required, or also in the context of question answering the full phrase may be required for appropriate response generation
  - on the other hand, chunking can add additional capability beyond simple part-of-speech tagging, without significant additional computational cost
- some terminology : chunk is what interests us (e.g., noun phrases), and a chink is what does not interest us, so we leave the chinks out of our chunks

- implementation -
  - regular expression search patterns is one method to implement chunking. there are identifiable patterns associated to different types of phrases, e.g., a noun phrase can be identified as a sequence of article—adjective—adjective—noun. this is readily implemented. but, for robust chunking, the number of rules required grows quickly, and requires update and maintenance.
  - fortunately, packages such as NLTK include built-in and highly reliable chunkers, so the rules and patterns do not need to be re-invented
  - there is a common notation for chunk identification, that supports straightforward extraction of desired chunk types from NLTK or similar chunk packages. the notation includes : I (token inside chunk), O (token outside any chunk - chink), and B (token begins a new chunk). following this pattern supports extraction of target chunk types from tagged tet returned from shallow parsers
- utilizing -
  - named entity recognition (NER) - chunking for noun phrases serves an important 1st step in downselecting to candidate phrases for subsequent evaluation
  - numerous applications - topic modeling, classification, clustering, semantic analysis, , and even just word-clouding can all benefit from the ability to selectively types of phrases to include / exclude in analyses. shallow parsing - chunking - provides expeditious means to implement those selections
- full parsing
  - method 1 - constituency parser - successively breaks a sentence into smaller and smaller segments, retaining identification of the hierarchical structure of the sentence and each of the subsequent sub-phrases. A constituency parser provides a graphical depiction of the relations via a directed acyclic graph - in which the root is sentence, interior nodes are phrase tags, the penultimate nodes are part-of-speech tags, and the word in the sentence are the leaf nodes. The edges are unlabeled.
  - method 2 - a dependency parser identifies a central theme and identifies the relationship of several sub-elements to that central function. As an example, the subject and object of a verb can be associated by their joint dependency on the shared verb. In this case, the dependency parser practices zen buddhism in that there is no subject without object and there is no object without subject. A dependency parser provides

a graphical depiction of the relations via a directed acyclic graph - in which the root is typically the primary verb, the edges are labeled with the connecting function, and the words comprise the nodes.

- implementation -
  - one method is the CYK algorithm. the sentence structure is traversed searching for a match to pre-identified grammar rules. when a sequence of POS-tagged words matches a grammar rule, then that provides tag for that portion of the sentence. the token sets are scanned at successively broader scope until (hopefully) the entire sentence has been contained within known grammar structures. python package pyStatParser includes a constituency parser implementing a variant of the CYK method.
  - established parsers available include : MST and MALT parsers for dependency parsing, and the Stanford and Link grammar parsers for constituency parsing.

9 Unit 10 : Semantic Analysis: Semantic Relatedness

- Semantic similarity - an interesting concept since objects can be similar in many different ways, across many different dimensions. Semantic similarity can consider similarity on the basis of : words; sense; text; taxonomy; frame; and context.
- word similarity - there are (generally) two approaches used to measure word similarity : statistical and structural. (further described here after). from a practical view, the uses of word similarity include : search expansion, plagiarism detection, identification of change/differences between texts and-or usages. This last
  - statistical approaches include :
    - PPMI - ”positive point-wise mutual information”
      - a measure of the relative distinction between dependent and independent word pairs. If a word is more likely to appear in combination more likely than it is to appear without the associated word, then there is a high dependency. A way to measure this is with PMI. positive PMI indicates that words in a pair are related. The way to express PMI is

$$PMI(X,Y) = log_2 \frac{P(x,y)}{P(x)P(y)} \tag{1}$$

Since this function ranges from -infinity to infinity, and the concept of negative infinity is a bit odd in the context of word associations (what does -infinity mean for word associations?), the function is typically trimmed at 0 and values less than zero are assigned as zero values. This measure is then called PPMI (for positive PMI)

* vector semantics LSA (”latent semantic analysis”) - aka distributional semantics - inspired by J.R.Firth ”a word is characterized by the company it keeps”. the items here below provide some metrics for assessing the semantical similarities / dissimilarities. From a practical point of view, for reasonably sized corpora, the vectors / matrices associated to term semantics are very sparse. for that reason, a method employed to manage these sparse matrices more efficiently is latent semantic analysis ... which is essentially an application of singular value decomposition of the matrix which produces a smaller, denser set of information for characterizing the corpora semantically.
  · term-document matrix - counts occurrences of words in documents - document similarity can be determined by the vector similarity of their term-document vectors; similarly, words similarity can be measured by the vector similarity also of term-document matrix. for these purposes, word similarity vectors are the transpose of the document similarity vectors. Often, tf-idf (term frequency - inverse document frequency) are preferred over straight term frequency counts; this provides normalization for documents of differing length
  · term-context matrix - rather than measure frequency of individual words, a term context matrix assembles a matrix of word counts and their near-neighbors ($\pm\frac{n}{2}$-words, n-grams)
* cosine similarity - a way to measure the distance between 2 vectors. since words are expressed as vectors, a vector similarity measure is appropriate. the cosine similarity is the vector dot product and measures the Euclidean distance two vectors

$$\cos(\mathbf{t}, \mathbf{e}) = \frac{\mathbf{t}\mathbf{e}}{\|\mathbf{t}\|\|\mathbf{e}\|} = \frac{\sum_{i=1}^{n} \mathbf{t}_i \mathbf{e}_i}{\sqrt{\sum_{i=1}^{n} (\mathbf{t}_i)^2} \sqrt{\sum_{i=1}^{n} (\mathbf{e}_i)^2}} \tag{2}$$

  - ∗ Jaccard distance - another measure of similarity / dissimilarity; Jaccard distance is the ration of the number of elements in two sets divided by the number of elements in either set, i.e., the intersection of the sets divided by the union of the sets. mathematically :

    $$J(X, Y) = \frac{|X \cap Y|}{|X \cup Y|} * 100 \tag{3}$$

    the percentage from the above relation indicates the degree of similarity : 100% = identical, 50% indicates the two sets share half of their members; 0% indicates no commonality
  - ∗ Hellinger distance - another potential similarity mesaurement. Hellinger distance measures the distance between two probability distributions.; it is considered a distriutional analog to the Euclidean distance. If the Hellinger distance is small (for identified features), then the two distributions are not similar. Or, those features (for small distance) are not differentiating features for the compared distributions. Mathematically, the Hellinger distance is expressed as :

    $$h(p, q) = \frac{1}{\sqrt{2}} ||\sqrt{p} - \sqrt{q}||_2 \tag{4}$$

- • structural approaches - assess similarity based on comparison of how words are parsed (i.e., with similar modifiers)
  - ∗ ontological distance - using an ontology (word tree hierarchical structure) to determine the distance between terms. essentially, ascending and descending through a semantic hierarchy to count the number of steps from one term to the next. a significant challenge associated this approach, is that an ontology that encompasses all of the significant terms must exist to support the measurement. constructing a reasonably complete and sufficiently accurate ontology for specialty domains is a time consuming, manual (especially from sknowledgeable SMEs), and somewhat error prone process.

10 Unit 11 : Semantic Analysis: Document Clustering

– centroid clustering - typically this defaults to k-means clustering. essentially, using the bag of words characterization of a document as the multidimensional characterization of that document, and the centroid (either mean or median) as the defining characteristic of the document. within a collection of documents, the centroid of each bog-of-words can be represented in a multi-dimensional space. within the k-means clustering method, each cluster is defined by its location in the vector space; the centroid of each cluster is found in an iterative approach : each document is randomly assigned to a cluster, the centroid of that cluster is calculated, and then each document is re-evaluated to determine in which cluster it now belongs (distance); documents are re-assigned, as appropriate, and then cluster centroids are re-calculated based on updated document assignments. and continuing, until no new adjustments of documents to clusters is identified. Some shortcomings of k-means clustering : the number of clusters is pre-determined. Since this an unsupervised learning approach, there is no predetermination of the best number of clusters. Typically, a range of number of clusters is evaluated, and metrics (such as perplexity) can be used to guide an appropriate number of clusters. In addition, since the initial assignment of document to cluster is random, and this is not a deterministic solution, different initial conditions can produce different final clustering definitions. To understand the potential clustering solutions, often multiple iterations of clusterings are conducted, with randomized initial states, to appreciate the potential differences in clustering that are produced from differing initial conditions.

- hierarchical clustering - as the name implies, the clustering produced by these methods is hierarchical in nature ... high level clusters encompass large numbers of class members, and each class is sequentially sub-divided into narrower and narrower subsets, with identification remaining with each higher level class. Two methods of hierarchical clustering are identified in below bullets. Interestingly, both of these methods were developed by the same team - Kaufman and Rousseeuw - in 1990, and there are the most widely used methods.
  - Agglomerative Nesting (AGNES) - approach is to begin at the leaf node (single document) level, identify next similar documents, and continue to build until all documents are included in an overall single hierarchy. the approach to agglomerative clustering, is to use variance measure to associate documents that produces least increase in within cluster variance with each successive association.
  - Divisive Analysis (DIANA) - opposite of AGNES, consider all documents in single highest level class and sequentially divide each class

until each document finds its appropriate leaf node position. The approach to achieve divisive clustering is to identify the furthest outlier within the cluster and assign to a new cluster. This is kind of like my sister is clearly an outlier in relation to me and my brothers, and then my youngest brother is the oddball in comparison to me and my next-to-youngest brother. (Just adding a little levity, in case you are still reading this far down the document ... and you have read about twenty of these already !) This is also an iterative method, as each document is re-assessed to determine to which cluster it is associated after each additional division is decided.

- Ward's minimum variance method - used typically in implementation of agglomerative clustering. the basic concept is to
  - * determine the mean of each cluster; calculate distance between each object in a cluster and the distance to the cluster's mean; square these distances; sum all of the sums of squares; this sum of sum of squares is the variance index for the current state. to determine next agglomeration, all possible combination of clusters for one agglomeration are determined; the smallest increase in the sum of sum of squares is chosen for the next agglomeration.
- this is a computationally expensive method; it generally results in a acceptable results
- The variance index can be expressed as [1] :

$$\sum_i \sum_j \sum_k |X_{ijk} - \bar{x}_{i\cdot k}|^2 \tag{5}$$

11 Unit 12 : Semantic Analysis: Document Classification

– in contrast to clustering, document classification is : requires the classifications to be present in advance; it is a supervised learning method; requires pre-labeled groupings. considerations in document classification include
  - subject based vs. descriptor based
  - binary vs. multiclass
  - text vs. document classification
  - overall landscape classification

[1] https://onlinecourses.science.psu.edu/stat505/node/146/

- content based classification are typically implemented using some form of multinomial naïve Bayes. implementation of naïve Bayes exist in sklearn package in python. Bayes theorem is a statement of conditional probability:

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}, \tag{6}$$

  where, in this case, we consider that A is a potential target category for a text B. For the naïve Bayes aspect, the consideration is that all words are independent (rather than being dependent on a phrase or n-gram association). In this way, a text B can consist of either an individual word or a phrase or set of words to be (potentially) associated to a category A. The implementation of naïve Bayes is straightforward calculation of the frequency counts of term occurrences and classification occurrences, and then direct calculation of the probability of each term for each category, and then assignment of term to category that provides highest probability. In practice, occurrence of $P(B|A)$ may not exist within the target corpus, so that some statistical smoothing is necessary to manage to zero occurrence cases. This is implemented via Laplace smoothing (or similar methods).
- In addition to naïve Bayes, Support Vector Machine (SVM) classifications are also implemented in NLP. SVM identifies the hyper-plane(s) that provide best separation between classifications.

12 Unit 13 : Semantic Analysis: Topic Modeling

- topic modeling falls into a few broad categories : canonical, organic, entity-centric
- canonical - match a pre-established list of topics within the identified domain (i.e., supervised). the canonical definitions come from authoritative sources (e.g. Library of Congress established document classifications)
- organic - discover the naturally occurring topics within the corpus (i.e., unsupervised) - organic is currently in favor with NLP teams, likely associated to the fact that it does not rely on authoritative source or training sets ... it is straightforward to implement with a sufficiently large corpus and some identified business case
- organic methods :
  - Latent Dirichlet Allocation (LDA) - uses probability distributions to guide the assignment of words to topics. the concept is that a topic is as-

sociated to groups of words that share high co-occurrences among different documents. different topics are afforded shared words, provided those shared words occur frequently across topics. mathematically, the LDA relation is expressed as [2]:

$$P(k|d) = P(k|z) \cdot P(z|d), \tag{7}$$

where $P(k|z)$ is the proportion of assignments to topic z among all docs that possess work k and $P(z|d)$ is the proportion of words in document d assigned to topic z.. The implementation of LDA is an iterative algorithm (in similar fashion to k-means clustering) in that words are assigned to topics randomly at 1st iteration, and then iteratively re-assigned based on assessment of improvement in the probabilities determined from the above relation. LDA is reasonably economical to implement (from a computational expense point of view) and is considered very accurate. LDA is most often used when moderate number of topics is desired (i.e., less than 50). LDA is a very favored algorithm in current topic modeling projects.

- Non-negative Matrix Factorization (NMF) - a variant of LDA in which the parameters have been defined in such a way as to produce a very limited number of topics. NMF is most useful in situations where document length is short. It is less computationally expensive than LDA, works well without significant re-tuning, and has the reputation to perform better on small corpora than either LSA or LDA.

13 Unit 14 : Semantic Analysis - Sentiment & Rhetoric

- basic sentiment analysis : identify positive or negative sentiment associated to a block of text. sentiment scoring is typically conducted on a [-1, +1] scale (-1 = negative, +1 = positive sentiment). there are two main approaches to sentiment scoring : supervised learning and unsupervised lexical KB approach.
- the machine learning approach has the advantages : it is quick to implement if there is a large training set available and there is no need to develop a specific coded vocabulary to initiate the analysis. On the other hand : the

[2] Bergamaschi & Po 2015

results of this approach are not transparent for subsequent interpretation, and the analysis is not any more finely tuned than was the available training data set.

- in terms of implementation of machine learning, the pipeline is similar to other supervised learning approaches : identify the training set, normalize texts (methods of normalization vary depending on intended purpose), extract feature vectors from the normalized text, train the classifier (typically binary and likely SVM), evaluate results, decide what additional improvements are needed
- in contrast to machine learning approach, the lexical approach uses existing sentiment vocabularies to support the analysis. some advantages of the lexical approach are that no training data is required (already provided with the lexicon) and the results of the analysis are explainable by association to the sentiment assigned in the reference lexicon. some downsides include : there is a need to maintain a tagged lexicon, and the difficulty to manage an up-to-date lexicon in the face of changing vocabulary and even changing nuances/sentiments associated to existing vocabulary. choices for established lexicons include :
    - AFINN - Affective Lexicon by Finn Nielsen, which includes approx 2500 clues
    - Liu's lexicon - 6800 clues
    - MPQA (Multi-Perspective Question Answering) - subjectivity lexicon: 8,222 clues
    - SentiWordNet: labels all 100k+ WordNet synsets !! (autogenerated)
    - VADER (Valence Aware Dictionary for sEntiment Reasoning): 7,500 clues
    - Pattern library lexicon (2,912 clues, mostly adjectives, with IDs from WordNet, valences hand coded or inferred from a nearest neighbor)
    - custom - as you choose
- in terms of implementation of this lexicon based approach, the pipeline is similar to other NLP analysis methods : identify the chosen semantically tagged lexicon, normalize texts (methods of normalization vary depending on intended purpose), extract feature vectors from the normalized text, implement some scoring method for evaluation, evaluate results, decide what additional improvements are needed

The thing about finishing a story is that finishing is really only the beginning.

William Herring