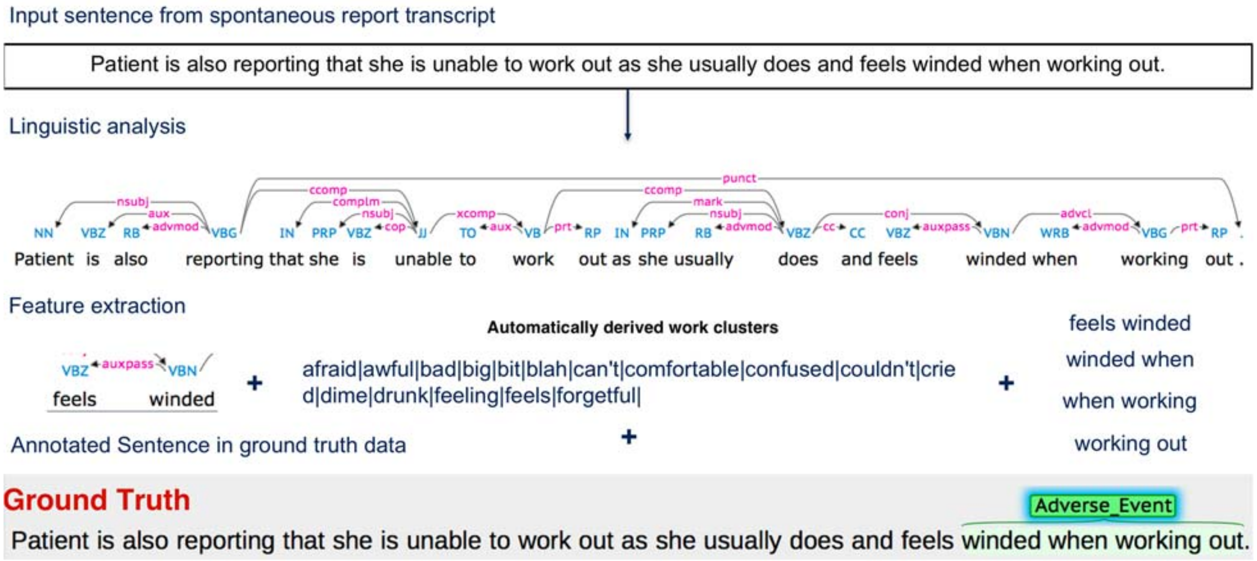
Sorting through the Safety Data Haystack: Using Machine Learning to Identify Individual Case Safety Reports (ICSRs) in Social-Digital Media

Drug Safety

Shaun Comfort, Sujan Perera, Zoe Hudson, Darren Dorrell, Shawman Meireis, Meenakshi Nagarajan, Cartic Ramakrishnan and Jennifer Fine

Corresponding Author:
Shaun Comfort, MD
Genentech - A Member of the Roche Group
South San Francisco, CA 94080
comforts@gene.com

Electronic Supplementary Material 2 Background on machine learning approaches to natural language processing


Input sentence from spontaneous report transcript

Patient is also reporting that she is unable to work out as she usually does and feels winded when working out.

Linguistic analysis

Feature extraction

Annotated Sentence in ground truth data

Ground Truth

Patient is also reporting that she is unable to work out as she usually does and feels winded when working out.
nsubj
aux
advmod
ccomp
complm
nsubj
cop
xcomp
aux
prt
ccomp
mark
nsubj
advmod
punct
cc
conj
auxpass
advcl
advmod
prt
NN VBZ RB VBG IN PRP VBZ JJ TO VB RP IN PRP RB VBZ CC VBZ VBN WRB VBG RP .
Patient is also reporting that she is unable to work out as she usually does and feels winded when working out .
VBZ auxpass VBN
feels winded
+
Automatically derived work clusters
afraid|awful|bad|big|bit|blah|can't|comfortable|confused|couldn't|cried|dime|drunk|feeling|feels|forgetful|
+
feels winded
winded when
when working
working out
+
Adverse_Event

Figure 1 Linguistic pre-processing and feature extraction used to train an Adverse Event annotator under the Knit platform

Figure 1 shows a sample sentence that contains the mention of a single adverse event expressed by the phrase “winded when working out”. The figure shows how this sentence is passed through a linguistic preprocessing step that parses the input sentence to generate a representation of the grammatical structure of the input sentence. Such a representation, known as a dependency parse tree expresses the grammatical role based, named dependencies between words in an input sentence. Using the result of such pre-processing as a base, we develop feature extractors that extract key features from such sentences. Such extracted features are then combined to train a model that can learn the characteristics of adverse event mentions in text, expressed as a function of the presence of such feature combinations. This function is represented as a model which can then be used to detect mentions of adverse events in previously unseen text.

Consider the three types of features shown here. The first, represented as a fragment of the parse tree connecting the words ‘feels’ and ‘winded’. Such a feature function is represented as a template that is designed to match pairs of words that a connected by a syntactic arc labeled ‘auxpass’. When such a template is applied to the word ‘winded’, which in the ground truth is marked as an adverse event, the template yields the pattern ‘feels’←auxpass←’winded’.

In addition to syntactic features we use word features derived via corpus-wide clustering methods. One such feature, shown in Figure 1, contains collections of words relating to how a patient experiencing an adverse even might feel (e.g. afraid, awful etc.). A feature function applied to words is then defined based on which cluster of words each word belongs.

We also utilize features that capture sequential co-occurrence of words. Such feature templates are defined based on pairs of consecutive words and each such pair is treated as a feature.

It should be noted that these are only representative examples of feature functions that we use. The Knit platform supports the construction and use of a wide variety of feature functions.

In addition to the capability of training annotators, the KniT platform has infrastructure that supports the construction of a wide variety of classifiers. For the experiments presented within this paper we use a Support Vector Machine (SVM) classifier to train a model that can classify documents as valid or invalid ICSRs. SVMs belong to a class of supervised classification algorithms known as maximum margin classifiers. We used the LIBSVM (https://www.csie.ntu.edu.tw/~cjlin/libsvm/) implementation of SVMs. This implementation was coupled with the feature extraction technology to train a binary classifier that can use these features to classify documents as valid or invalid ICSRs.