Exposing Predictive Analytics through Natural Language

Jeroen van Grondelle[1], Christina Unger[2], and Frank Smit

[1] HU University of Applied Sciences Utrecht
jeroen.vangrondelle@hu.nl
[2] CITEC, Bielefeld University
cunger@cit-ec.uni-bielefeld.de

Abstract

Data processing applications often have non-experts as audience, which has led to an increasing interest into verbalizing data into natural language. However, verbalizations often focus on observed or predicted facts, while predictive technologies in addition generate information such as probabilities and confidence scores. This gives rise to data that goes beyond facts alone and that also has to be communicated to its users in order to enable a correct interpretation of the predictions. In this paper we explore the natural language needed when applications offer predictive analytics technology to their users, and demonstrate how to modularly implement the grammars needed to verbalize common aspects of predictions.

1 Introduction

In order to allow non-expert users to access data, a lot of study has gone into applying natural language generation for the verbalization of knowledge- and databases [1], recently in the context of the Semantic Web [2]. However, data services often include some form of reasoning engines and predictive analytics. As a result, the dialog between user and system includes inferred data and predictions. With that, a number of aspects enter the dialog beyond simple fact assertion. For example, predictive techniques create statements about specific future expectations, and often include additional information on, e.g., confidence, probability and provenance of the predicted facts. These aspects have subtle semantics that non-statistically trained people are not familiar with, but that do impact which expectations are justified based on the predictions. Sharing these formal attributes with the users of these predictive services is important to ensure a correct interpretation of the results [3].

In order to enhance the accessibility as well as comprehensibility of predictive data for non-expert users, this paper proposes a number of natural language verbalizations for different kinds of predictive data. In a proof of concept, different machine learning algorithms are applied to flight delay data. Based on the LT^3 framework [6], modular grammars are developed which are used to generate verbalizations of flight delay predictions. These grammars comprise both a domain grammar module for verbalizations of flight delays, and a generic grammar module covering predictive aspects, that could also be coupled with any other domain.

2 Verbalizing Predictive Data

Predictive analytics techniques discover and analyse patterns in data about historical behaviour and events, and use those to predict likely future events and behaviour, for example the revenue

a customer is going to generate, expected house prices, or flight delays. The underlying technologies often make some form of generalisation of the data they train on, in order to predict on unseen datapoints. The basic data of predictive analytics are thus future facts, for instance the time that the departure of a flight will be delayed. In its most basic form, a fact like The departure of the flight is delayed by 10 minutes can be turned into a prediction of a future fact by turning it into simple future tense: The departure of the flight will be delayed by 10 minutes.

In addition, predictive techniques can produce data on aspects that enrich a prediction with further information that describes the prediction or the way it was reached, the specific semantics of which depends on the type of predictive model that is employed. In this paper we focus on the following aspects that we identified as common across a number of predictive techniques:

Interval In case of continuous numerical data, a classifier can be used to predict a discretized class, i.e. an interval of values. Also, a regression technique could provide a margin, e.g. based on its error rate, in order to offer an interval prediction instead of an exact value, thereby encoding its inability to make a precise prediction.

Intervals can be verbalized either by means of their limits (e.g. The departure of the flight will be delayed between 10 and 20 minutes) or by means of a name or description (e.g. The departure of the flight will be slightly delayed).

Probability Some predictive techniques, such as Naive Bayes classification, inherently produce probabilistic predictions, i.e. they predict the probability of a certain fact materializing. That fact not materialising in a certain individual case by no means disqualifies the prediction; instead more detailed statistical analyses across multiple instances is needed to check if the observed frequency of the fact can be reconciled with the predicted probability.

Similar to intervals, the probability of a prediction can be expressed as an exact value (e.g. There is a chance of 83% that the departure of the flight will be delayed by 10 minutes) or as a description of the probability range expressed by an adjective or adverb (e.g. There is a high chance that the departure of the flight will be delayed by 10 minutes or The departure of the flight will most likely be delayed by 10 minutes).

In addition, instead of trying to verbalize a probability in the context of a single occurrence, the very nature of probability suggests the alternative of verbalizing the effect of that probability across occurrences. This, again, can be done in an exact way, as a percentage of known cases (e.g. The departure of similar flights is delayed by 10 minutes in 83 % of the cases), or in a descriptive way codifying the percentage by adverbs such as often (e.g. The departure of similar flights is often delayed for 10 minutes).

Confidence In addition, a measure of confidence can be associated along with a prediction, that specifies to which extent the prediction can be trusted. Its exact implementation depends on the type of classifier or regressor. In case no confidence measure is implemented, the prediction accuracy could serve as an indicator for confidence. In contrast to a probabilistic prediction, an individual prediction with an attached confidence value must me interpreted as false if it does not prescribe the actual future behaviour.

In order to convey the semantics of a confidence associated with a prediction, it is important to clearly distinguish it from the prediction itself and its probability. This is easiest done by not verbalizing the predicted future event as a fact, but rather express the potentiality of the event using different modalities, indicated by modal markers such as certainly and maybe or modal auxiliaries (e.g. The departure of the flight might be delayed for

10 minutes, or modal constructions such as It seems like the departure of the flight will be delayed for 10 minutes), or using speaker attitudes, e.g. by means of constructions such as I am convinced that..., I believe that..., I am undecided whether....

Some predictive techniques may use more than one of these classes of information in their predictions, i.e. predict a probability for a variable to be in a certain interval. The forms for the different kinds of predictive data can then be combined if the predictive technology produces results that have more than one of these properties associated to it.

In order to create a modular grammar resource for predictive verbalizations that facilitates the porting of those verbalizations across domains and languages, we use the LT^3 framework introduced in [6]. Most importantly, its architecture decouples domain aspects (e.g. flight delays or customer revenues) from task aspects (e.g. predictions and dialog aspects). This allows for an easy coupling of the predictive task with any domain of data. Moreover, its architecture relies on an automatic generation of grammars based on lexical resources that require less effort and linguistic expertise to create.

3 Case: Predicting and Communicating Flight Delays

In this section we instantiate LT^3 for the use case of predicting flight delays. To this end, we realize a prototype for training common machine learning algorithms on flight delay data, predicting the delay of single flight instances, and building verbalizations that express the resulting prediction in natural language.

The prototype is realized in Python[1] and is available at `http://bitbucket.org/jcgronde/exposingpredictiveanalytics`. It imports predictive techniques from the Scikit-learn Toolkit[2] and builds on Grammatical Framework[3] (GF) [5] for natural language generation. The prototype slices the data, picking a (either specified or random) data point for prediction and all other data points for training, producing a prediction and a corresponding GF representation as output. The latter is piped through the GF shell for linearization in two languages: English and Dutch.

We applied the system to the flight records[4] provided by RITA[5], which capture information about U.S. domestic flights from 1987 on, including (among others) date information, the origin and destination of the flight, its carrier, the scheduled departure and arrival times, the departure and arrival delay as well as delay causes. As accurate prediction is not a goal in itself for our research, we have considered only a subset of the data, in particular all those flights from 2008 that originate from one of three airports (John F. Kennedy International, Dallas Love Field, Honolulu International) and have specified delay causes, amounting to 53,817 flight instances. In general, the delays peak at delays up to 40 minutes, leaving a large tail of longer delays.

For training and prediction we have taken into account the following columns (disregarding all others): month, day of month, day of week, scheduled departure and arrival time, unique carrier code, origin and destination airport, travelled distance, departure and arrival delay (in minutes), as well as delay causes. In addition, we add columns that pool values of the month, day of week, distance and delay columns into coarse classes, in particular NO for delays of 0 minutes, LOW for delays from 1 to 20 minutes, MEDIUM for delays between 21 and 40 minutes, and HIGH for delays greater than 40 minutes.

[1] Python, Version 2.7, `http://python.org/`
[2] Scikit-learn Toolkit, Version 0.14, `http://scikit-learn.org/`
[3] Grammatical Framework, Version 3.5, `http://grammaticalframework.org/`
[4] `http://stat-computing.org/dataexpo/2009/the-data.html`
[5] `http://www.transtats.bts.gov/OT_Delay/OT_DelayCause1.asp`

The departure and arrival delay are the target columns to be predicted, the others are used as input for the prediction. Predictions are made with a Gaussian Naive Bayes classifier, a Linear SVM classifier, a decision tree classifier, and a K-nearest neighbors regressor with $k = 3$. For arrival and departure delays as prediction targets, the regressor predicts an exact delay value, while the classifiers predict one of the above mentioned coarse classes (NO, LOW, MEDIUM, HIGH).

3.1 Lexical Resources and Domain Grammar

For generating verbalizations, we exploit the LT^3 framework [6], which builds on a pipeline that generates grammar resources from conceptualizations and according declarative lexicalizations. To apply this framework, we therefore construct a domain ontology that captures flights together with the properties captured by the dataset, in particular their origin and destination airport, their carrier, the distance they span, their scheduled departure and arrival time, as well as the departure and arrival delays.

The lexical representation of those domain aspects are captured in a *lemon*[6] [4] lexicon. *lemon* is a model for the declarative specification of multilingual, machine-readable lexica in RDF that capture syntactic and semantic aspects of lexical items relative to some ontology. The meaning of a lexical item is given by reference to an ontology element, i.e. a class, property or individual, thereby ensuring a clean separation between the ontological and lexical layer.

Both the ontology and lexica in English and Dutch are available at Bitbucket: `http://bitbucket.org/chru/lexica`.

The domain ontology and the corresponding lexica then serve as input for *lemongrass*[7], a grammar generation script that automatically constructs a GF grammar, mapping the ontology to an abstract syntax and each lexicon to a concrete syntax.

Since the lexicon only contains domain-specific expressions (mainly nouns, verbs and adjective), but no domain-independent functional expressions, such as determiners, negation and coordination, also the resulting domain grammar is incomplete for the purposes of a full-fledged dialog. It is thus combined with a core grammar module[8] containing domain-independent expressions as well as clause building constructions.

This combined grammar now allows to express domain facts, e.g. the fact that the departure of a flight is delayed for 10 minutes. The part of the domain-independent core grammar module that contains numerals and units allows for expressing either exact values (10 minutes) or more vague descriptions (a few minutes or slightly). 1–3 show the corresponding verbalizations in English and Dutch.

1. The departure is delayed for 10 minutes.
   De vlucht vertrekt met een vertraging van 10 minuten.

2. The departure is delayed for a few minutes.
   De vlucht vertrekt met een vertraging van enkele minuten.

3. The departure is slightly delayed.
   Het vertrek is licht vertraagd.

Similarly, the grammar module for numericals allows to express intervals (e.g. between 10 and 20 minutes) as well as interval limits, for example:

[6] `http://lemon-model.net`
[7] `http://bitbucket.org/chru/lemongrass`
[8] Available for English and Dutch at `http://bitbucket.org/chru/grammars`.

4. The departure is delayed for at least 40 minutes.
   De vlucht vertrekt met een vertraging van tenminste 40 minuten.

Since the verbalization of exact and descriptive numerical values is contained in a domain-independent grammar module, it can be used in each domain that makes numerical statements, as well as in a task grammar module, e.g. for verbalizing exact or descriptive numerical probabilities, as shown below.

3.2 Generating Verbalizations of Predictive Data

Verbalization aspects of predictive analytics are captured in a domain-independent task grammar module[9], which, for now, was hand-crafted. It captures functions for building predictions from domain statements, as well as functions for attaching probability and confidence information to such predictions.

The GF representation d of a domain fact can now be embedded in a task-specific prediction function `prediction`, turning a domain fact into a simple prediction of a future fact by changing the domain verbalization into future tense, as shown in the following GF representation with corresponding verbalizations.

5. `predict` d
   The departure will be delayed for 10 minutes.
   De vlucht zal met een vertraging van 10 minuten vertrekken.

Analogously, any other domain fact can be embedded in the prediction, in order to yield descriptive verbalizations such as The departure will be delayed for a few minutes or The departure will be slightly delayed, and verbalizations specifying interval limits, for example The departure will be delayed for at least 40 minutes.

In addition to plain predictions of future facts, the task grammar defines functions that add probabilistic information to predictions p of the form `predict` d (with d any domain statement), either using exact values or descriptions, as in the following examples.

6. `probability_exact` p `(value_float_unit 83.0 Percent)`
   There is a chance of 83.0 percent that the departure will be delayed for 10 minutes.
   Er is een kans van 83.0 procent dat de vlucht met een vertraging von 10 minuten vertrekken zal.

7. `probability_descr` p `High`
   There is a high chance that the departure will be delayed for 10 minutes.
   Er is een hoge kans dat de vlucht met een vertraging von 10 minuten vertrekken zal.

7 uses a descriptive measure (`Low`, `Medium`, or `High`) that can be verbalized either as an adjective (in English slight, moderate and high, in Dutch licht, redelijk and hoog) or as an adjective modifier (in English slightly, moderately and significantly, in Dutch licht, redelijk and flink).

Similarly, the function `confidence` adds confidence information to a prediction p, expressing a confidence measure by means of attitudes, as in the following example.

8. `attitude` p `Expect`
   I expect that the departure will be delayed for 10 minutes.
   Ik verwacht dat de vlucht met een vertraging von 10 minuten vertrekken zal.

[9] Available for English, Dutch and German at `http://bitbucket.org/chru/grammars`.

4 Future Work

In this paper, we have proposed generic verbalizations of common information that is relevant for predictions, in particular intervals, probabilities and confidence measures. This, however, does not yet cover all aspects and information that predictive techniques offer. In order to base decisions on predictions, it would, for example, be necessary to also verbalize provenance information, such as explanations of the produced prediction in terms of the predicition model, or information about the general accuracy of a predictor. Another aspect that can enrich verbalizations are expectations, e.g. conveying whether delays are expected for a particular airport or carrier, or whether are they surprising. Similarly, we could verbalize conditional predictions that explicitly name factors that often change and thus might still alter the prediction (e.g. If the weather conditions do not change, the departure of the flight will not be delayed). Finally, alternative forms such as exact numerical vs descriptive verbalizations could be used to address different audiences, e.g. having different expertise levels.

In addition to an extension of the verbalized aspects, future work will include an evaluation of the performance of the proposed verbalizations, investigating whether non-statistically trained users understand the sentences and their consequences, and whether users experience the actual outcomes to be in line with the expectations they got from the verbalized predictions.

Acknowledgments

This work was partially funded by the EU project PortDial (FP7-296170).

References

[1] Ion Androutsopoulos. Natural language interfaces to databases – an introduction. *Journal of Natural Language Engineering*, 1:29–81, 1995.

[2] Nadjet Bouayad-Agha, Gerard Casamayor, and Leo Wanner. Natural language generation in the context of the semantic web. *Semantic Web*, 2013.

[3] Anthony Jameson. Understanding and Dealing With Usability Side Effects of Intelligent Processing. *AI Magazine*, 30(1994):23–40, 2009.

[4] John McCrae, Guadelupe Aguado de Cea, Paul Buitelaar, Philipp Cimiano, Thierry Declerck, Asuncion Gomez-Perez, Jorge Garcia, Laura Hollink, Elena Montiel-Ponsoda, and Dennis Spohr. Interchanging lexical resources on the semantic web. *Language Resources and Evaluation*, 46(4):701–719, 2012.

[5] Aarne Ranta. *Grammatical Framework: Programming with Multilingual Grammars*. CSLI Publications, 2011.

[6] Jeroen Van Grondelle and Christina Unger. A Three-Dimensional Paradigm for Conceptually Scoped Language Technology. In *Towards the Multilingual Semantic Web*. Springer, 2014.