

Automatic Tweet Generation From Traffic Incident Data

Khoa Tran

School of Computing Science
Simon Fraser University
Burnaby, BC, CANADA
khoa_tran@sfu.ca

Fred Popowich

School of Computing Science
Simon Fraser University
Burnaby, BC, CANADA
popowich@sfu.ca

Abstract

We examine the use of traffic information with other knowledge sources to automatically generate natural language tweets similar to those created by humans. We consider how different forms of information can be combined to provide tweets customized to a particular location and/or specific user. Our approach is based on data-driven natural language generation (NLG) techniques using corpora containing examples of natural language tweets. It specifically draws upon semantic data and knowledge developed and used in the web based Connected Vehicles and Smart Transportation system. We introduce an alignment model, generation model and location-based user model which will together support location-relevant information delivery. We provide examples of our system output and discuss evaluation issues with generated tweets.

1 Introduction

Traffic congestion continues to be a major problem in large cities around the world, and a source of frustration for commuters, commercial drivers, tourists, and even occasional drivers. Current efforts to reduce congestion and frustration often involve providing road users with real-time traffic information to help estimate travel time accurately, resulting in better route planning and travel decisions (Tseng et al., 2013). The different approaches to deliver traffic information include radio, smart navigation devices and social networks.

Information from radio and social networks is delivered as messages, which consist primarily of natural language. When delivered on smart navigation devices, information is presented with

colour and icons on interactive maps; for example, congested road segments are usually in red while clear road segments are in green ¹.

Text and audio messages associated with radio and social network channels are mainly human-generated, requiring time and effort. The information sources for these messages primarily uses the same data used as smart navigation devices in conjunction with camera images, eye-witness reports and other sources, which collectively require substantial effort and time. Although several social network channels may use computer programs (i.e., “bots”) to generate messages automatically from a data source, these messages are constructed using strict templates which appear to users as cold, unnatural, distant and unreliable.

2 Our Approach

We look at the role of natural language generation (NLG) in the context of a system that automatically generate messages about traffic incidents. Our approach is based on data-driven NLG techniques where corpora containing examples of natural language tweets are used to train the model to generate natural language texts. It draws upon semantic data and knowledge developed and used in the web based Connected Vehicles and Smart Transportation (CVST) system (Tizghadam and Leon-Garcia, 2015). We introduce an alignment model, generation model and location-based user model which together support location-relevant information delivery.

We design a traffic notification system having a location-based user model to predict a user’s routes and deliver real-time notifications if traffic incidents occur. Figure 1 shows the design of our proposed system. The GPS location of a user is

¹Google Maps uses this colour code as described in <https://support.google.com/maps/answer/3092439?hl=en&rd=1>

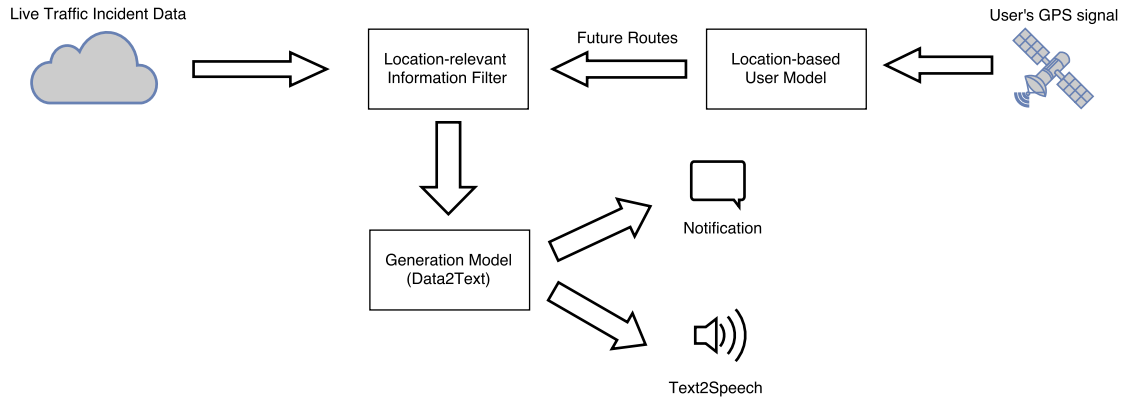


Figure 1: The traffic notification system that notifies location-relevant traffic information for road users.

processed through the location-based user model. It predicts a ranked list of routes and destinations the user could take. Concurrently, a live stream of traffic incidents is collected and forwarded to the location-relevant information filter. This component applies a location filter on the traffic incident data based on the predicted user’s routes and destinations. The output is the data scenarios of traffic incidents that happen on or nearby the routes the user may take. Next, the generation model composes short messages describing nearby traffic incidents. These messages are sent to users as textual or speech notifications using a text-to-speech system.

We construct our corpus from two data-sets from the CVST APIs ². The first data-set is a collection of 13,667 tweets mentioning traffic incidents in the greater Toronto area of Canada. The second data-set consists of 27,795 records concerning road incidents in greater Toronto. Using incident times and locations from the two data-sets, we are able to match tweets with the road incidents to construct a corpus of road incidents with their corresponding tweets. We also explore other traffic related data-sets that can be used to train our system. However, using such data is restricted as discussed in Section 4.1. On the other hand, for each road incident in our constructed corpus, we are able to collect more than one tweet from different users mentioning the event. Utilizing these human-generated texts ensures better output quality for our NLG system.

Using the constructed corpus, we apply an existing semantic alignment model (Section 4.2) to learn the semantic correspondences between data

records and their textual descriptions in the tweets. Then, we apply a model for concept-to-text generation (Section 4.3) to generate tweets about traffic incidents from given records. However, our system’s output is not limited to tweet generation. Output can be personalized, for example, as a virtual assistant, to generate traffic notifications for users based on their driving routines incorporating daily routes, departure and arrival times, and specific locations. Previous work on capturing users’ locations and route prediction (Section 4.4) can be applied to select only the potentially user-interested traffic information and deliver it to the drivers.

The evaluation of automatically generated tweets can be approached from several perspectives. Evaluation in the context of the task outlined in Figure 1 can involve human subjects, looking at metrics such as the usefulness of tweets (using rating criteria like those in rating the helpfulness of reviews or comments), and the quality of tweets (involving fluency and readability). Detailed human evaluation of the tweets is beyond the scope of the current research. Our plan is to focus on automated techniques in the evaluation of the automatically generated tweets, given that we have a gold-standard of human generated data. To evaluate our models, we build on previous evaluation techniques such as BLEU and METEOR (Konstas, 2014).

3 Related work

Recent work in automatic tweet generation focuses on the tasks of automatic text summarization and topic classification. Lloret and Palomar (2013) present a framework that automatically

²<http://portal.cvst.ca>

generates twitter headlines for journal articles using text summarization approaches. Analogously, by applying different text processing techniques including content grouping, topic classification and text summarization, Lofi and Krestel (2012) develop a system that generates tweets from government documents. Krokos and Samet (2014) also utilize several sentiment analysis and classification methods in their approach to automatically discover and generate hashtags for tweets that do not have user-generated hashtags. On the other hand, Sidhaye and Cheung (2015) use different metrics and statistics to show that most tweets cannot be re-constructed from the original articles that they reference; concluding that applying extractive summarization methods to generate indicative tweets could be a limitation.

Our work focuses on another aspect where tweets are constructed from structured data, and in our case the data can come from a real-time web application. Our generation task is also known as data-to-text, concept-to-text or linguistic description of data generation. The domain of our NLG work is novel with respect to the previous work in different domains including weather forecasts (Ramos-Soto et al., 2015a), educational reports (Bontcheva and Wilks, 2004; Ramos-Soto et al., 2015b) and clinical reports (Portet et al., 2009).

Although the results from previous work are promising and some proven to be better than human-generated content, they still have limitations. Most current approaches are based on very specific rules or grammars. Therefore, adapting these systems to a different data-set or domain usually requires re-designing the entire system again. However, data-driven techniques are applicable to different domains (Liang et al., 2009; Angeli et al., 2010; Kim and Mooney, 2010; Konstas, 2014). These approaches define probabilistic models that can be trained to learn the patterns and hidden alignments between data and text, thereby avoiding the construction of rules and grammars that require domain-specific knowledge.

We focus on a generation system that can apply to different types of traffic data-sets (road closures, road incidents, traffic flow, etc.). We use an existing alignment model (Liang et al., 2009) to learn the semantic correspondences between traffic data and its textual description. Konstas (2014)’s concept-to-text generation approach is used for the automatic generation of tweets to be

ultimately incorporated into a real-time system.

Overall, we chose our data-driven approach since it is location independent; different cities have different kinds of traffic data, information road closures, road incidents, road conditions each with different kinds of data structures. We can handle different data-sets without changing the model structure, incorporating it into an end-to-end system with surface realisation and content planning in one model.

4 The Task

A data entry \mathbf{d} consists of a set of records $\mathbf{r} = \{r_1, r_2, \dots, r_n\}$. Each record is described with a record type $r_i.t$, $1 \leq i \leq |\mathbf{r}|$, and a set of fields \mathbf{f} . Each field $f_j \in \mathbf{f}$, $1 \leq j \leq |\mathbf{f}|$, has a field type $f_j.t$ and a value $f_j.v$. A scenario in the training corpus is a pair of (\mathbf{d}, \mathbf{w}) where \mathbf{w} is the text describing the data entry \mathbf{d} . Our goal is to train a model that represents the hidden alignments between data entry \mathbf{d} and the observed text \mathbf{w} in the training corpus. Then, the trained model that captures the alignment is used to generate text \mathbf{g} from a new entry \mathbf{d} not contained in the training corpus.

4.1 Dataset

There are various types of traffic-related data including traffic flow, traffic incidents, road constructions and road closures. Such data is usually available through different map and road navigation APIs such as Tom Tom Traffic³, Google Maps⁴ and Bing Maps⁵ or government open data sources. Despite the wide availability of traffic-related data, most of the data are only useful for visualisation purposes since they lack the corresponding textual descriptions. A few of the data sources have a short description associated with each data entry such as Dublin City Council’s road works and maintenance⁶ and Bing Maps’ Traffic Incidents⁷. However, the text description is not sufficiently detailed to cover essential information in the data entry.

The CVST project has APIs for different traffic-related data-sets of greater Toronto area including

³<http://developer.tomtom.com/>

⁴<https://developers.google.com/maps/>

⁵<https://www.bingmapsportal.com/>

⁶<https://data.dublincity.ie/dataset/roads-maintenance-annual-works-programme>

⁷<https://msdn.microsoft.com/en-us/library/hh441726.aspx?f=255&MSPPEror=-2147217396>

traffic cameras and sensors, road closures and incidents, public transportation and tweets. We use two data-sets from the CVST APIs, road incidents and twitter incidents, to construct our corpus. The road incidents data-set has details about traffic incidents such as time, location, type and reason. The twitter incidents data-set contains basic information about the incident and its related tweets.

By matching times and locations of records in the two data-sets, we construct a parallel corpus of traffic incidents with their related tweets. However, the times and locations from the two data-sets are not always exactly matched. Therefore, we allow errors when matching these values. We consider two incidents from two data-sets to be matched if:

- the events' locations are within 100 meters from each other,
- and the events' start times are within 90 minutes of each other

The data is collected from January 2015 to May 2016. There are 27,795 records in road incident data-set and 13,134 records with 13,667 tweets in the twitter incident data-set (some records have more than one associated tweets). After matching the two data-sets using the described rules, we have a corpus of 1,388 incidents and 2,829 tweets. The tweets are crawled from Twitter and are generated by both humans and machines.

4.2 The alignment model

Liang et al. (2009) introduce a hierarchical semi-Markov model to learn the correspondences between a world state and an unsegmented stream of text. Their approach is a generative process with three main components:

- Record choice: choose a sequence of records $r = (r_1, \dots, r_{|\mathbf{r}|})$ where each $r_i \in \mathbf{d}$ and has a record type $r_i.t$. The choice of consecutive records depends on their types.
- Field choice: for each chosen record r_i , select a sequence of fields $\mathbf{f}_i = (f_{i1}, \dots, f_{i|\mathbf{f}_i|})$ where each $f_{ij} \in \{1, \dots, m\}$.
- Word choice: for each chosen field f_{ij} , choose a number $c_{ij} > 0$ and generate a sequence of c_{ij} words.

Their record choice model is described as a Markov chain of records conditioned on record types. Their intention is to capture salience and coherence. Formally:

$$p(\mathbf{r} | \mathbf{d}) = \prod_{i=1}^{|\mathbf{r}|} p(r_{i.t} | r_{i-1.t}) \frac{1}{|\mathbf{s}(r_{i.t})|}$$

where $\mathbf{s}(r_{i.t})$ is the set of records in \mathbf{d} that has record type $r_{i.t}$ and $r_{0.t}$ is the START record type. Their model also includes a special NULL record type responsible for generating text that does not belong to any real record types. Analogously, field choice model is a Markov chain of fields conditioned on the choice of records:

$$p(\mathbf{f} | \mathbf{r}) = \prod_{i=1}^{|\mathbf{r}|} \prod_{j=1}^{|\mathbf{f}_j|} p(f_{ij} | f_{i(j-1)})$$

Two special fields — START and STOP — are also implemented to capture the transitions at the boundaries of the phrases. In addition, each record type has a NULL field aligned to words that refer to that record type in general. The final step of the process is the word choice model where words are generated from the choice of records and fields. Specifically, for each field f_{ij} , we generate a number of words c_{ij} , chosen uniformly. Then the words \mathbf{w} are generated conditioned on the field \mathbf{f} .

$$p(\mathbf{w} | \mathbf{r}, \mathbf{f}, \mathbf{c}, \mathbf{d}) = \prod_{k=1}^{|\mathbf{w}|} p_w(w_k | r(k).t_{f(k)}, r(k).v_{f(k)})$$

where $r(k)$ and $f(k)$ are record and field responsible for generating word w_k and $p_w(w_k | t, v)$ is the distribution of words given a field type t and field value v . Their model supports three different field types. Depending on the field types, Liang et al. 2009 define different methods for generating words:

- Integer type: generate the exact value, rounding up, rounding down and adding or subtracting unexplained noise ϵ_+ or ϵ_-
- String type: generate a word chosen uniformly from those in the field value
- Categorical type: maintain a separate multinomial distribution over words for each field value in the category.

G_{CS}	1. $S \rightarrow R(start)$	$[Pr = 1]$
	2. $R(r_i.t) \rightarrow FS(r_j, start) R(r_j.t)$	$[P(r_j.t r_i.t) \frac{1}{ S(r_j.t) }]$
	3. $R(r_i.t) \rightarrow FS(r_j, start)$	$[P(r_j.t r_i.t) \frac{1}{ S(r_j.t) }]$
	4. $FS(r, r.f_i) \rightarrow F(r, r.f_j)FS(r, r.f_j)$	$[P(f_j f_i)]$
	5. $FS(r, r.f_i) \rightarrow F(r, r.f_j)$	$[P(f_j f_i)]$
	6. $F(r, r.f) \rightarrow W(r, r.f)F(r, r.f)$	$[P(w w_{-1}, r, r.f)]$
	7. $F(r, r.f) \rightarrow W(r, r.f)$	$[P(w w_{-1}, r, r.f)]$
G_{SURF}	8. $W(r, r.f) \rightarrow \alpha$	$[P(\alpha r, r.f, f.t, f.v, f.t = cat, null)]$
	9. $W(r, r.f) \rightarrow gen(f.v)$	$[P(gen(f.v).mode r, r.f, f.t = int) \times P(f.v gen(f.v).mode)]$
	10. $W(r, r.f) \rightarrow gen_str(f.v, i)$	$[Pr = 1]$

Table 1: Grammar rules used for generation with their corresponding weights.

4.3 The generation model

Konstas (2014) recasts an earlier model (Liang et al., 2009) into a set of context-free grammar (CFG) rules. To capture word-to-word dependencies during the generation process, he added more rules to emit a chain of words, rather than words in isolation. Table 1 shows his defined grammar rules with their corresponding weights.

The first rule in the grammar expands from a start symbol S to a special START record $R(start)$. Then, the chain of two consecutive records, r_i and r_j is defined through rule (2) and (3). Their weight is the probability of emitting record r_j given record r_i and corresponds to the record choice model of Liang et al. (2009). Equivalently, rule (4) and (5) define the chain of two consecutive fields, f_i followed by f_j , and their weight corresponds to the field choice model. Rule (6) and (7) are added to specify the expansion of field F to a sequence of words W . Their weight is the bigram probability of the current word given its previous word, the current record and field. Finally, rules (8)-(10) are responsible for generating words. If the field type is categorical (denoted as *cat*) or NULL (denoted as *null*), rule (8) is applied to generate a single word α in the vocabulary of the training set. Its weight is the probability of seeing α , given the current record, field and the field type is *cat* or *null*. Rule (9) is applied if the field type is integer (denoted as *int*). $gen(f.v)$ is a function that accepts the field value (an integer) as its input and return an integer using one of the six methods described by Liang et al. (2009). The weight is a multinomial distribution over the six integer generation function choices, given the record field f , times $P(f.v | gen(f.v).mode)$, which is set to the geometric distribution of noise

ϵ_+ and ϵ_- , or to 1 otherwise (Konstas, 2014). Rule (10) is responsible for generating a word for string-type field. $gen_str(f.v, i)$ is a function that simply return the i^{th} word of the string in the field value $f.v$.

After defining the grammar rules, Konstas (2014) treats the generation problem as a parsing problem using the CFG rules. He uses a modified version of the CYK algorithm (Kasami, 1966; Younger, 1967) to find the best text w given a structured data entry d . His basic decoder is presented as a deductive proof system (Shieber et al., 1995) in Table 2. The decoding process works in a bottom-up fashion. It starts with choosing N — the length (number of words) of the output text. Konstas (2014) determines N using a simple linear regression model where features being record-field pairs in the data entry d . Then, for each position i in the output text, it searches for the best scoring item that spans from i to $i + 1$ (one single word). Next, items are visited and combined in order for larger spans until it reaches the goal item $[S, 0, N]$ — symbol S spans from position 0 to N .

The basic decoder always chooses the best scoring item during the parsing process. Konstas (2014) extends the basic decoder with the k-best decoder in which a list of k-best derivations will be kept for each item. The extension significantly improves the output quality by avoiding local optimums. He also intersects the grammar rules with a tri-gram language model and a dependency model to ensure fluency and grammaticality of the output text.

4.4 Location-based user model

There has been wide range of work on location-based user models, learning and predicting users' routes and destinations. These tasks involve some

Items:	$[A, i, j]$ $R(A \rightarrow B)$ $R(A \rightarrow BC)$
Axioms:	$[W, i, i + 1] : s \quad W \rightarrow g_{i+1}, g_{i+1} \in \{\alpha, \text{gen}(), \text{gen_str}()\}$
Inference rules:	(1) $\frac{R(A \rightarrow B):s[B, i, j]:s_1}{[A, i, j]:s \cdot s_1}$ (2) $\frac{R(A \rightarrow BC):s[B, i, k]:s_1[C, k, j]:s_2}{[A, i, j]:s \cdot s_1 \cdot s_2}$
Goal:	$[S, 0, N]$

Table 2: The basic decoder deductive system.

uncertainties. Much work relies on GPS signal data to identify a user’s location and may not be accurate. In addition, intended destinations are not always certain since they may be affected by factors such as weather, traffic, day of week, and time of day. Due to many uncertainties arising from the task, most systems build probabilistic models to identify and predict users’ locations. Marmasse and Schmandt (2000, 2002) apply pattern recognition techniques to learn users’ patterns of traveling and frequent destinations. These frequent locations can be added or removed manually by users. Then, each location is assigned to a to-do list which is displayed whenever users travel to this location. In Krumm et al. (2013)’s model, the map can be modelled as a directed graph where road intersections are vertices and road segments connecting these intersections are edges. A probabilistic model is used to rank the potential destinations based on the current trip (previous intersections users have passed). After collecting a list of candidate destinations and their probabilities, route probabilities are computed by summing all destination probabilities along the fastest routes. Therefore, a corpus of the driver’s regular routes is not necessary in this model. Simons et al. (2006) use a Hidden Markov Model (HMM) with the extended version where they consider factors such as day of week and time of travel in their prediction algorithms, while Liao et al. (2007) use a more complex HMM with the ability to infer the user’s mode of transportation.

5 Examples

Table 3 presents an example of input and output of the generation model with different settings. In the first setting, we train the weights of the grammar rules using the whole corpus. Next, we use the k-best decoder integrated with a tri-gram language model to generate the text given the input scenario.

We try different values of k (the number of k-best derivations kept for each item during the generation process). The generation system generates output 1a and output 1b for $k = 10$ and $k = 20$ respectively. We can try with larger values of k , however, it will affect the generation time, which becomes a factor if incorporated into a real time system. In the above example, instead of using the whole corpus, we use only tweets from a specific user to train the model. The chosen tweets in the second setting are generated by the user “680 NEWS Traffic” who has the majority of tweets in the corpus. We also try different k values ($k = 10$ and $k = 20$), however, the results are the same and presented in output 2.

Some essential records such as “Reference road” and “Reason” from the input scenario are not chosen by the generator for inclusion in the generated tweets in output 1a and output 2 respectively. On the other hand, extra information that the input does not cover is included arbitrarily such as “collision”, “the right lane” or “the left lane”. There are two main reasons for this behavior:

- inaccurate alignments between data and text: all the alignments are inferred from an unannotated corpus. A fully or partially annotated corpus will improve the accuracy of the alignment model.
- corpus: usually, the tweets contain more information than the structured data. This extra information can create noise in the training process, especially without supervision.

6 Conclusions and Future Work

The preliminary results for the data-driven approaches show that it is possible to generate real-time tweets for inclusion in a real-time traffic notification system, using techniques that are oth-

Input	Main road		Reference road	Lane	Stream
	Name 401	Direction Eastbound	Name Yonge St	Value collectors	Value collectors
	Condition	Reason	Incident type		
	Value Bad traffic	Value Disabled vehicle	Value Disabled vehicle		
Output 1a	collision: collision: #hwy401 eb express at yonge st				
Output 1b	401 collectors - right lane blocked with a collision.				
Output 2	eb 401 at yonge express, blocking the left lane				

Table 3: Example input and output of the generation system with different settings.

erwise applicable to different domains and data-sets. There are various types and sources of traffic-related data useful to drivers (e.g. traffic flow, road construction,...), and we have only scratched the surface of the issues concerning personalized tweets. Further evaluation is required and we will present the preliminary results from automatic evaluation during the workshop.

A high priority for the ongoing research is the content-preference model: some users desire certain information more than other information (e.g. reason of the accident, detour information, ...). A content-preference model can be integrated into the grammar to re-rank the generated sentences with the information users need.

Given the relatively constrained domain, we want to consider how template based models can be used with the data-driven approach introduced in this paper. A template approach requires different set of patterns and rules for each traffic data type, but integrating techniques involving semantic role labels (Lindberg et al., 2013) may assist in applying our approach to different data-sets and different locations.

Another aspect we need to consider to improve the system is how can we optimize it in terms of output quality and generation time. For output quality, considering the limitation described in section 5, we may want to get more data and potentially annotate parts of the data to get better alignment accuracy. In addition, applying different data pre-processing and normalizing techniques can also help clean up the data before training the model. To improve the generation time, we can apply an approximate search approach such as cube-pruning (Chiang, 2007).

Finally, we will evaluate the system using metrics such as BLEU and METEOR given that we have human-generated data. These two metrics are also used for evaluation in Konstas (2014)’s work.

However, the data we have collected is comprised of both human-generated and machine-generated texts. Therefore, we need to develop a technique to separate the two sets. One simple way is based on the Twitter username generating the tweets. In addition, we can also set up experiments comparing how different the results are when the system is trained with only human-generated texts and is trained with both sets.

Acknowledgements

The authors would like to thank the reviewers for detailed and constructive feedback. Comments and insights from Anoop Sarkar and Milan Tofiloski were also greatly appreciated in improving the quality of the paper. Special thanks to Ali Tizghadam for assistance in providing access to the necessary data. This research was supported in part by the Natural Sciences and Engineering Research Council of Canada.

References

- Gabor Angeli, Percy Liang, and Dan Klein. 2010. A simple domain-independent probabilistic approach to generation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP '10*, pages 502–512, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Kalina Bontcheva and Yorick Wilks, 2004. *Automatic Report Generation from Ontologies: The MIAKT Approach*, pages 324–335. Springer Berlin Heidelberg, Berlin, Heidelberg.
- David Chiang. 2007. Hierarchical phrase-based translation. *Comput. Linguist.*, 33(2):201–228, June.
- T. Kasami. 1966. An efficient recognition and syntax-analysis algorithm for context-free languages. Technical Report AF-CRL-65-758, Air Force Cambridge Research Laboratory, Bedford, Massachusetts.

- Joohyun Kim and Raymond J. Mooney. 2010. Generative alignment and semantic parsing for learning from ambiguous supervision. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, COLING '10, pages 543–551, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Ioannis Konstas. 2014. *Joint Models for Concept-to-Text Generation*. Ph.D. thesis, University of Edinburgh.
- Eric Krokos and Hanan Samet. 2014. A look into twitter hashtag discovery and generation. In *Proceedings of the 7th ACM SIGSPATIAL Workshop on Location-Based Social Networks (LBSN14)*, Dallas, TX, Nov.
- John Krumm, Robert Gruen, and Daniel Delling. 2013. From destination prediction to route prediction. *J. Locat. Based Serv.*, 7(2):98–120, June.
- Percy Liang, Michael I. Jordan, and Dan Klein. 2009. Learning semantic correspondences with less supervision. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - Volume 1*, ACL '09, pages 91–99, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Lin Liao, Donald J. Patterson, Dieter Fox, and Henry Kautz. 2007. Learning and inferring transportation routines. *Artificial Intelligence*, 171(5):311 – 331.
- David Lindberg, Fred Popowich, John Nesbit, and Phil Winne. 2013. Generating natural language questions to support learning on-line. *ENLG 2013*, pages 105–114.
- Elena Lloret and Manuel Palomar. 2013. Towards automatic tweet generation: A comparative study from the text summarization perspective in the journalism genre. *Expert Systems with Applications*, 40(16):6624 – 6630.
- Christoph Lofi and Ralf Krestel, 2012. *iParticipate: Automatic Tweet Generation from Local Government Data*, pages 295–298. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Natalia Marmasse and Chris Schmandt. 2000. Location-aware information delivery with commotion. In *Proceedings of the 2Nd International Symposium on Handheld and Ubiquitous Computing*, HUC '00, pages 157–171, London, UK, UK. Springer-Verlag.
- Natalia Marmasse and Chris Schmandt. 2002. A user-centered location model. *Personal Ubiquitous Comput.*, 6(5-6):318–321, January.
- Franois Portet, Ehud Reiter, Albert Gatt, Jim Hunter, Somayajulu Sripada, Yvonne Freer, and Cindy Sykes. 2009. Automatic generation of textual summaries from neonatal intensive care data. *Artificial Intelligence*, 173(7):789 – 816.
- A. Ramos-Soto, A. J. Bugarn, S. Barro, and J. Taboada. 2015a. Linguistic descriptions for automatic generation of textual short-term weather forecasts on real prediction data. *IEEE Transactions on Fuzzy Systems*, 23(1):44–57, Feb.
- A. Ramos-Soto, M. Lama, B. Viquez-Barreiros, A. Bugarn, and M. M. S. Barro. 2015b. Towards textual reporting in learning analytics dashboards. In *2015 IEEE 15th International Conference on Advanced Learning Technologies*, pages 260–264, July.
- Stuart M. Shieber, Yves Schabes, and Fernando C.N. Pereira. 1995. Computational linguistics and logic programming principles and implementation of deductive parsing. *The Journal of Logic Programming*, 24(1):3 – 36.
- Priya Sidhaye and Jackie Chi Kit Cheung. 2015. Indicative tweet generation: An extractive summarization problem? In Llus Mrquez, Chris Callison-Burch, Jian Su, Daniele Pighin, and Yuval Marton, editors, *EMNLP*, pages 138–147. The Association for Computational Linguistics.
- R. Simmons, B. Browning, Yilu Zhang, and V. Sadekar. 2006. Learning to predict driver route and destination intent. In *2006 IEEE Intelligent Transportation Systems Conference*, pages 127–132, Sept.
- Ali Tizghadam and Alberto Leon-Garcia. 2015. Application platform for smart transportation. In *Future Access Enablers of Ubiquitous and Intelligent Infrastructures*, pages 26–32. Springer.
- Yin-Yen Tseng, Jasper Knockaert, and Erik T. Verhoef. 2013. A revealed-preference study of behavioural impacts of real-time traffic information. *Transportation Research Part C: Emerging Technologies*, 30:196 – 209.
- Daniel H. Younger. 1967. Recognition and parsing of context-free languages in time n^3 . *Information and Control*, 10(2):189 – 208.