UPV-28-UNITO at SemEval-2019 Task 7: Exploiting Post's Nesting and Syntax Information for Rumor Stance Classification

Bilal Ghanem[1], **Alessandra Teresa Cignarella**[1,2],
Cristina Bosco[2], **Paolo Rosso**[1], **Francisco Rangel**[1,3]
1. PRHLT Research Center, Universitat Politècnica de València
2. Dipartimento di Informatica, Università degli Studi di Torino
3. Autoritas Consulting
bigha@doctor.upv.es, cigna@di.unito.it,
prosso@dsic.upv.es, bosco@di.unito.it, francisco.rangel@autoritas.es

Abstract

In the present paper we describe the UPV-28-UNITO system's submission to the RumorEval 2019 shared task. The approach we applied for addressing both the subtasks of the contest exploits both classical machine learning algorithms and word embeddings, and it is based on diverse groups of features: stylistic, lexical, emotional, sentiment, meta-structural and Twitter-based. A novel set of features that take advantage of the syntactic information in texts is moreover introduced in the paper.

1 Introduction

The problem of rumor detection lately is attracting considerable attention, also considering the very fast diffusion of information that features social media platforms. In particular rumors are facilitated by large users' communities, where also expert journalists are unable to keep up with the huge volume of online generated information and to decide whether a news is a hoax (Procter et al., 2013; Webb et al., 2016; Zubiaga et al., 2018).

Rumour stance classification is the task that intends to classify the type of contribution to the rumours expressed by different posts of a same thread (Qazvinian et al., 2011) according to a set of given categories: supporting, denying, querying or simply commenting on the rumour. For instance, referring to Twitter, once a tweet that introduces a rumour is detected (the "source tweet"), all the tweets having a reply relationship with it, (i.e. being part of the same thread), are collected to be classified.

Our participation to this task is mainly focused on the investigation of linguistic features of social media language that can be used as cues for detecting rumors[1].

2 Related work

The RumorEval 2019 shared task involves two tasks: Task A (rumour stance classification) and Task B (verification).

Stance Detection (SD) consists in automatically determining whether the author of a text is in favour, against, or neutral towards a given target, i.e. statement, event, person or organization, and it is generally indicated as TARGET-SPECIFIC STANCE CLASSIFICATION (Mohammad et al., 2016).

Another type of stance classification, more general-purpose, is the OPEN STANCE CLASSIFICATION task, usually indicated with the acronym SDQC, by referring to the four categories exploited for indicating the attitude of a message with respect to the rumour: Support (S), Deny (D), Query (Q) and Comment (C) (Aker et al., 2017). Target-specific stance classification is especially suitable for analyses about a specific product or political actor, being the target given as already extracted, e.g. from conversational cues. On this regard several shared tasks have been organized in recent years: see for instance SemEval-2016 Task 6 (Mohammad et al., 2017) considering six commonly known targets in the United States, and StanceCat at IberEval-2017 on stance and gender detection in tweets on the matter of the Independence of Catalonia (Taulé et al., 2017). On the other hand, the open stance classification, (i.e. the task addressd in this paper), is more suitable in

[1] Source code is available on GitHub: https://github.com/bilalghanem/UPV-28-UNITO

Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), pages 1125–1131
Minneapolis, Minnesota, USA, June 6–7, 2019. ©2019 Association for Computational Linguistics

classifying emerging news or novel contexts, such as working with online media or streaming news analysis.

Provided that attitudes around a claim can act as proxies for its veracity, and not only of its controversiality, it is reasonable to consider the application of SDQC techniques for accomplishing rumour analysis tasks. A first shared task, concerning SDQC applied to rumor detection, has been organized at SemEval-2017, i.e RumorEval 2017 (Derczynski et al., 2017). Furthermore, several research works have analyzed the open issue of the impact of rumors in social media (Resnick et al., 2014; Zubiaga et al., 2015, 2018), for instance exploiting linguistic features (Ghanem et al., 2018). Such a kind of approaches may be also found in works which deal with the problems of Fake News Detection (Ciampaglia et al., 2015; Hanselowski et al., 2018).

Furthermore, a rumor is defined as a "circulating story of questionable veracity, which is apparently credible but hard to verify, and produces sufficient scepticism and/or anxiety so as to motivate finding out the actual truth" (Zubiaga et al., 2015).

Concerning veracity identification, increasingly advanced systems and annotation schemas have been developed to support the analysis of rumour veracity and misinformation in text (Qazvinian et al., 2011; Kumar and Geethakumari, 2014; Zhang et al., 2015).

3 Description of the task

The RumorEval task is articulated in the following sub-tasks: **Task A** (open stance classification – SDQC) is a multi-class classification for determining whether a message is a "support", a "deny", a "query" or a "comment" wrt the original post; **Task B** (verification) is a binary classification for predicting the veracity of a given rumour into "true" or "false" and according to a confidence value in the range of 0-1.

3.1 Training and Test Data

The RumourEval 2019 corpus contains a total of 8,529 English posts, namely 6,702 from Twitter and 1,827 from Reddit.

The portion of data from Twitter has been built by combining the RumorEval 2017 training and development datasets (Derczynski et al., 2017), and includes **5,568** tweets: 325 source tweets (grouped into eight overall topics such as Charlie Hebdo attack, Ottawa shooting, Germanwings crash...), and 5,243 discussion tweets collected in their threads.

The dataset from Reddit, which has been instead newly released this year, is composed by **1,134** posts: 40 source posts and 1,094 collected in their threads.

	Training	Test
Twitter	5,568	1,066
Reddit	1,134	761
Total	6,702	1,827

Table 1: Training and test data distribution.

All data have been split in training and test set with a proportion of approximately $80\% - 20\%$ (see Table 1).

4 UPV-28-UNITO Submission

The approach and the features selection we applied is the same for both tasks and is based on a set of manual features described in Section 4.1. We built moreover another set of features (i.e. second-level features) extracted by using the manual features together with features based on word embeddings (see Section 4.2 for a detailed description). For modeling the features distribution with respect to each thread, we used for task B the same features as in task A. Then, in both tasks, we fed the features to a classical machine learning classifier.

4.1 Manual Features

For enhancing the selection of features, we investigated the impact of diverse groups of them: emotional, sentiment, lexical, stylistic, meta-structural and Twitter-based. Furthermore, we introduced a novel set of syntax-based features.

Emotional Features - We exploited several emotional resources in order to build features for our system. Three lexica: (a) **EmoSenticNet**, a lexicon that assigns six WordNet Affect emotion labels to SenticNet concepts (Poria et al., 2013); (b) the **NRC Emotion Lexicon**, a list of English words and their associations with eight basic emotions (anger, fear, anticipation, trust, surprise, sadness, joy, and disgust) and two sentiments (negative and positive) (Mohammad and Turney, 2010);

and (c) **SentiSense**, an easily scalable concept-based affective lexicon for Sentiment Analysis (De Albornoz et al., 2012). We also exploited two tools: (d) **Empath**, a tool that can generate and validate new lexical categories on demand from a small set of seed terms (Fast et al., 2016); and (e) **LIWC** a text analysis dictionary that counts words in psychologically meaningful categories (Pennebaker et al., 2001).

Sentiment Features - Our sentiment features were modeled exploiting sentiment resources such as: (a) **SentiStrength**, a sentiment strength detection program which uses a lexical approach that exploits a list of sentiment-related terms (Thelwall et al., 2010); (b) **AFINN**, a list of English words rated for valence with an integer between minus five (negative) and plus five (positive) (Nielsen, 2011); (c) **SentiWordNet**, a lexical resource in which each WordNet synset is associated to three numerical scores, describing how objective, positive, and negative the terms contained in the synset are (Esuli and Sebastiani, 2007); (d) **EffectWordNet**, a lexicon about how opinions are expressed towards events, which have positive or negative effects on entities (+/-effect events) (Choi and Wiebe, 2014); (e) **SenticNet**, a publicly available resource for opinion mining built exploiting Semantic Web techniques (Cambria et al., 2014); and (f) the **Hu&Liu** opinion lexicon[2].

Lexical Features - Various lexical features already explored in similar Sentiment Analysis tasks were employed: (a) the presence of **Bad Sexual Words**, a list extracted from the work of Frenda et al. (2018); (b) the presence of **Cue Words** related to the following categories: *belief, denial, doubt, fake, knowledge, negation, question, report* (Bahuleyan and Vechtomova, 2017); the categories *an, asm, asf, qas, cds* of the multilingual hate lexicon with words to hurt **HurtLex** (Bassignana et al., 2018); (d) the presence of **Linguistic Words** related to the categories of *assertives, bias, fatives, implicatives, hedges, linguistic words, report verbs*; (e) the presence of specific categories present in **LIWC**: *sexual, certain, cause, swear, negate, ipron, they, she, he, you, we, I.* (Pennebaker et al., 2001).

Stylistic Features - We employed canonical stylistic features, already thoroughly explored in Sentiment Analysis tasks and already proven useful in multiple domains: (a) the count of **question marks**; (b) the count of **exclamation marks**; (c) **length** of a sentence; (d) the **uppercase ratio**; (e) the count of consecutive **characters** and **letters**[3] (f) and the presence of **URLs**.

In addition to the above-listed, common features exploited in Sentiment Analysis tasks, in this work we introduce two novel sets of features: (1) **Problem-specific features** (considering the fact that the dataset is composed by Twitter data and Reddit data) and (2) **Syntactical features**.

Meta-structural features - Since training and test data are from Twitter and Reddit both, we explored meta-structural features suitable for data coming from both platforms: (a) the **count of favourites/likes**, in which we have two different value distribution (Twitter vs. Reddit), so we normalized them in a range 0-100; (b) the **creation time** of a post, encoded in seconds; (c) the **count of replies**; and (d) the **level**, i.e. the degree of “nestedness” of the post in the thread.

Twitter-only Features - Because of the duplicitous nature of the RumorEval 2019 dataset (Twitter and Reddit), some of the several features, already thoroughly used in Sentiment Analysis tasks and based on Twitter metadata, could not be used in this task[4]. As follows: (a) the presence of **hashtags**; (b) the presence of **mentions**; (c) the count of **retweets**. And also some user-based features: (d) whether the user is **verified** or not; (f) the count of **followers**; (g) the count of **listed** (i.e. the number of public lists of which this user is a member of); (h) the count of **statuses**; (i) the count of **friends** (i.e. the number of users that one account is following); (l) the count of **favourites**.

Syntactic Features - In our system some feature has been also modeled by referring to syntactic information involved in texts (Saif et al., 2016). After having parsed[5] the dataset in the *Universal De-*

[2] http://www.cs.uic.edu/liub/FBS

[3] We considered 2 or more consecutive characters, and 3 or more consecutive letters.

[4] For the instances from Reddit, that did not have a representation of one of the following features, the empty values has been filled with a weighted average of the values obtained by other similar instances.

[5] The parsing system we applied is UDPipe, available at: https://pypi.org/project/ufal.udpipe/

pendency[6] format, thus obtaining a set of syntactic "dependency relations" (*deprel*), we were able to exploit: (a) the **ratio of negation** dependencies compared to all the other relations; (b) the Bag of Relations (BoR_all) considering all the *deprels* attached to **all the tokens**; (c) the Bag of Relations (BoR_list) considering all the *deprels* attached to the tokens belonging to a selected **list of words** (from the lists already made explicit in the paragraph "Lexical Features" in Section 4.1); and finally (d) Bag of Relations (BoR_verbs) considering all the *deprels* attached to all the **verbs**, thus fully exploiting morpho-synctactic knowledge.

4.2 Second-level Features

For the second-level features, we employed (a) the cosine similarity of one instance wrt its parents and (b) information of the tree structure of a thread, exploiting its "nesting" and depth from the source tweet.

Similarity with Parents - In this feature, we used the cosine similarity to measure the similarity between each post with its parents. The parents of a reply are the (A) direct upper-level post and (B) the source post in the thread (see Figure 1).

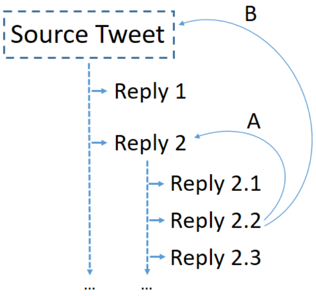

Figure 1: An example for reply 2.2 parents.

We extracted the cosine similarity in A and B by using the manual features' final vector and words embeddings average vectors of the posts; the words embeddings average vector for a post is extracted by averaging the embeddings of the post's words[7].

SDQC Depth-based Clusters - We built level-based stance clusters from the posts. For each stance class (SDQC), we extracted all the belonging posts that correspond to one of the four classes and we computed the average value of the feature vectors (as one unique cluster). Since we have four main stances, this process ended with four main clusters. For the feature extraction, we measured the cosine similarity for each post wrt these four clusters. As done in the previous feature described above, we built these clusters by using both the manual features' vectors and word embeddings' vectors of the posts, so each stance cluster is represented in two ways. In these four main clusters, we didn't consider the nesting of the posts in the thread.

Also, we obtained the same clusters but instead of averaging all the posts that correspond to a stance, we considered the nesting of the posts in the thread. We split the nesting of the threads into five groups: posts with depth one, two, three, four, five or larger. For each of these levels, we extracted four SDQC clusters (depth-based). For instance, if a post occurs in depth two, we measured the cosine similarity between this post and 1) the four main SDQC clusters[8], 2) the four depth-based SDQC clusters two.

Concerning task B, we modeled the distribution of the features used for task A. For each thread we did the following:

1. We counted how many posts in the thread correspond to each of the stances.
2. We extracted the averaged features' vectors for each stance's posts in the thread.
3. We extracted the standard deviation for each stance's posts in the thread.

5 Experiments

We tested different machine learning classifiers in each task performing 10-fold cross-validation. The results showed that the Logistic Regression (LR) produces the highest scores. For tuning the classifier, we used the Grid Search method. The parameters of the LR are: $C = 61.5$, $penalty = L2$, and since the dataset is not balanced, we used different weights for the classes as COMMENT =

[6] The *de facto* standard for the representation of syntactical knowledge in the NLP community: https://universaldependencies.org/

[7] We used the pre-trained Google News word embeddings in our system: https://code.google.com/archive/p/word2vec/

[8] Four features using the manual features, and another four using the words embeddings.

0.10, DENY = 0.35, SUPPORT = 0.20 and QUERY = 0.35. We conducted an ablation test on the features employed in task A in order to investigate their importance in the classification process. Table 2 presents the ablation test results as well as the system performance using 10-fold cross-validation.

SET	FEATURE	M-F1
A	All features	54.9
B	A - Emotional features	54.5
C	A - Sentiment features	54.7
D	A - Lexical features	53.6
E	A - Syntactic features	54.7
F	A - Stylistic features	50.1
G	A - Meta-structural features	54.5
H	A - Twitter-only features	54.9
I	A - Cosine similarity with parents	55.3
I.1	I using only manual features	54.9
I.2	I using only words embeddings	54.9
J	A - SDQC depth-based clusters	47.7
J.1	J using only manual features	53.3
J.2	J using only words embeddings	51.1
K	A-(C+E+I)	55.6
L	A-(B+C+E+G)	55.7
M	A-(B+C+E+G+I.2)	55.9

Table 2: Ablation test.

Provided that the organizers allowed two submissions for the final evaluation, on both tasks we used all the features (set A) in the first submission and set M for the second submission. In Table 3 we present the final scores achieved on both tasks.

	MACRO-F1	RMSE
Task A	48.95	–
Task B	19.96	82.64

Table 3: Final results.

6 Error Analysis

A manual error analysis allow us to see which categories and posts turned out to be the most difficult to be dealt with our system. We found out that SUPPORT was misclassified 114 times, DENY 92 times, QUERY 44 times, and COMMENT 57 times. Therefore, SUPPORT seems to be the hardest category to be correctly classified.

Table 4 reports the detailed confusion matrix of predicted vs. gold labels and shows that the most of errors are related to the category SUPPORT (in the gold dataset) and COMMENT (in our runs), while any error involves the more contrasting classes (e.g. SUPPORT and DENY). By better investigating the gold test set, it should be moreover observed that several semantically empty messages of the test set have been marked using some class, while our system marks them as COMMENT, i.e. selecting the more frequent class when a clear indication of the content is lacking.

		PREDICTED			
		S	D	Q	C
GOLD	S	–	0	13	101
	D	1	–	6	85
	Q	5	1	–	38
	C	5	17	35	–

Table 4: Confusion matrix of errors.

7 Conclusion

In this paper we presented an overview of the UPV-28-UNITO participation for *SemEval 2019 Task 7 - Determining Rumour Veracity and Support for Rumours*.

We submitted two different runs in the detection of rumor stance classification (Task A) and veracity classification (Task B) in English messages retrieved from Twitter and Reddit both. Our approach was based on emotional, sentiment, lexical, stylistic, meta-structural and Twitter-based features. Furthermore, we introduced two novel sets of features, i.e. *syntactical* and *depth-based* features, which proved to be successful for the task of rumor stance classification, where our system ranked as 5th (out of 26) and, according to the RMSE score, we ranked 6th in Task B for veracity classification. Since the two latter groups of features produced an interesting contribution to the score for Task A, but they were fairly neutral in Task B, we will follow this trail and try to inquire more on these aspects in our future work.

Acknowledgments

The work of Cristina Bosco was partially funded by Progetto di Ateneo/CSP 2016 (*Immigrants, Hate and Prejudice in Social Media*, S1618L2BOSC01).

References

Ahmet Aker, Leon Derczynski, and Kalina Bontcheva. 2017. Simple open stance classification for rumour analysis. *arXiv preprint arXiv:1708.05286*.

Hareesh Bahuleyan and Olga Vechtomova. 2017. UWaterloo at SemEval-2017 Task 8: Detecting

Stance Towards Rumours with Topic Independent Features. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 461–464.

Elisa Bassignana, Valerio Basile, and Viviana Patti. 2018. Hurtlex: A Multilingual Lexicon of Words to Hurt. In *5th Italian Conference on Computational Linguistics, CLiC-it 2018*, volume 2253, pages 1–6. CEUR-WS.

Erik Cambria, Daniel Olsher, and Dheeraj Rajagopal. 2014. SenticNet 3: a Common and Common-sense Knowledge Base for Cognition-driven Sentiment Analysis. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*.

Yoonjung Choi and Janyce Wiebe. 2014. +/-effectwordnet: Sense-level Lexicon Acquisition for Opinion Inference. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1181–1191.

Giovanni Luca Ciampaglia, Prashant Shiralkar, Luis M Rocha, Johan Bollen, Filippo Menczer, and Alessandro Flammini. 2015. Computational Fact Checking from Knowledge Networks. *PloS one*, 10(6).

Jorge Carrillo De Albornoz, Laura Plaza, and Pablo Gervás. 2012. SentiSense: An Easily Scalable Concept-based Affective Lexicon for Sentiment Analysis. In *LREC*, pages 3562–3567.

Leon Derczynski, Kalina Bontcheva, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Arkaitz Zubiaga. 2017. SemEval-2017 Task 8: RumourEval: Determining Rumour Veracity and Support for Rumours. *arXiv preprint arXiv:1704.05972*.

Andrea Esuli and Fabrizio Sebastiani. 2007. SentiWordNet: a High-coverage Lexical Resource for Opinion Mining. *Evaluation*, 17:1–26.

Ethan Fast, Binbin Chen, and Michael S. Bernstein. 2016. Empath: Understanding Topic Signals in Large-scale Text. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 4647–4657. ACM.

Simona Frenda, Bilal Ghanem, and Manuel Montes-y Gómez. 2018. Exploration of Misogyny in Spanish and English Tweets. In *3rd Workshop on Evaluation of Human Language Technologies for Iberian Languages (IberEval 2018)*, volume 2150, pages 260–267. CEUR-WS.

Bilal Ghanem, Paolo Rosso, and Francisco Rangel. 2018. Stance Detection in Fake News A Combined Feature Representation. In *Proceedings of the 1st Workshop on Fact Extraction and VERification (FEVER)*, pages 66–71.

Andreas Hanselowski, Avinesh PVS, Benjamin Schiller, Felix Caspelherr, Debanjan Chaudhuri, Christian M Meyer, and Iryna Gurevych. 2018. A Retrospective Analysis of the Fake News Challenge Stance Detection Task. *arXiv preprint arXiv:1806.05180*.

KP Krishna Kumar and G Geethakumari. 2014. Detecting misinformation in online social networks using cognitive psychology. *Human-centric Computing and Information Sciences*, 4(1):14.

Saif M. Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. 2016. SemEval-2016 Task 6: Detecting Stance in Tweets. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 31–41.

Saif M. Mohammad, Parinaz Sobhani, and Svetlana Kiritchenko. 2017. Stance and Sentiment in Tweets. *ACM Transactions on Internet Technology*, 17(3):26:1–26:23.

Saif M. Mohammad and Peter D. Turney. 2010. Emotions Evoked by Common Words and Phrases: Using Mechanical Turk to Create an Emotion Lexicon. In *Proceedings of the NAACL HLT 2010*, pages 26–34. ACL.

Finn Årup Nielsen. 2011. A new ANEW: Evaluation of a Word List for Sentiment Analysis in Microblogs. *arXiv preprint arXiv:1103.2903*.

James W. Pennebaker, Martha E. Francis, and Roger J. Booth. 2001. Linguistic Inquiry and Word Count: LIWC 2001. *Mahway: Lawrence Erlbaum Associates*, 71.

Soujanya Poria, Alexander Gelbukh, Amir Hussain, Newton Howard, Dipankar Das, and Sivaji Bandyopadhyay. 2013. Enhanced SenticNet with Affective Labels for Concept-based Opinion Mining. *IEEE Intelligent Systems*, 28(2):31–38.

Rob Procter, Farida Vis, and Alex Voss. 2013. Reading the Riots on Twitter: Methodological Innovation for the Analysis of Big Data. *International Journal of Social Research Methodology*, 16(3):197–214.

Vahed Qazvinian, Emily Rosengren, Dragomir R Radev, and Qiaozhu Mei. 2011. Rumor has it: Identifying Misinformation in Microblogs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1589–1599. ACL.

Paul Resnick, Samuel Carton, Souneil Park, Yuncheng Shen, and Nicole Zeffer. 2014. Rumorlens: A System for Analyzing the Impact of Rumors and Corrections in Social Media. In *Proceedings of the Computational Journalism Conference*.

Hassan Saif, Yulan He, Miriam Fernandez, and Harith Alani. 2016. Contextual Semantics for Sentiment Analysis of Twitter. *Information Processing & Management*, 52(1):5–19.

Mariona Taulé, Maria Antònia Martí, Francisco M. Rangel Pardo, Paolo Rosso, Cristina Bosco, and Viviana Patti. 2017. Overview of the Task on Stance and Gender Detection in Tweets on Catalan Independence. In *Proceedings of the 2nd Workshop on Evaluation of Human Language Technologies for Iberian Languages (IberEval 2017)*, volume 1881, pages 157–177. CEUR-WS.org.

Mike Thelwall, Kevan Buckley, Georgios Paltoglou, Di Cai, and Arvid Kappas. 2010. Sentiment Strength Detection in Short Informal Text. *Journal of the American Society for Information Science and Technology*, 61(12):2544–2558.

Helena Webb, Pete Burnap, Rob Procter, Omer Rana, Bernd Carsten Stahl, et al. 2016. Digital Wildfires: Propagation, Verification, Regulation, and Responsible Innovation. *ACM Transactions on Information Systems (TOIS)*, 34(3):15.

Qiao Zhang, Shuiyuan Zhang, Jian Dong, Jinhua Xiong, and Xueqi Cheng. 2015. Automatic detection of rumor on social network. In *Natural Language Processing and Chinese Computing*, pages 113–122. Springer.

Arkaitz Zubiaga, Ahmet Aker, Kalina Bontcheva, Maria Liakata, and Rob Procter. 2018. Detection and Resolution of Rumours in Social Media: A Survey. *ACM Computing Surveys (CSUR)*, 51(2):32.

Arkaitz Zubiaga, Maria Liakata, Rob Procter, Kalina Bontcheva, and Peter Tolmie. 2015. Towards Detecting Rumours in Social Media. In *AAAI Workshop: AI for Cities*.