WarwickDCS: From Phrase-Based to Target-Specific Sentiment Recognition

Richard Townsend
University of Warwick
richard@sentimentron.co.uk

Adam Tsakalidis
University of Warwick
A.Tsakalidis@warwick.ac.uk

Yiwei Zhou
University of Warwick
Yiwei.Zhou@warwick.ac.uk

Bo Wang
University of Warwick
Bo.Wang@warwick.ac.uk

Maria Liakata
University of Warwick
M.Liakata@warwick.ac.uk

Arkaitz Zubiaga
University of Warwick
A.Zubiaga@warwick.ac.uk

Alexandra Cristea
University of Warwick
acristea@dcs.warwick.ac.uk

Rob Procter
University of Warwick
rob.procter@warwick.ac.uk

Abstract

We present and evaluate several hybrid systems for sentiment identification for Twitter, both at the phrase and document (tweet) level. Our approach has been to use a novel combination of lexica, traditional NLP and deep learning features. We also analyse techniques based on syntactic parsing and token-based association to handle topic specific sentiment in subtask C. Our strategy has been to identify subphrases relevant to the designated topic/target and assign sentiment according to our subtask A classifier. Our submitted subtask A classifier ranked fourth in the SemEval official results while our BASELINE and μPARSE classifiers for subtask C would have ranked second.

1 Introduction

Twitter holds great potential for analyses in the social sciences both due to its explosive popularity, increasing accessibility to large amounts of data and its dynamic nature. For sentiment analysis on twitter the best performing approaches (Mohammad et al., 2013; Zhu et al., 2014) have used a set of rich lexical features. However, the development of lexica can be time consuming and is not always suitable when shifting between domains, which examine new topics and user populations (Thelwall and Buckley, 2013). Excitingly, the state of the art has recently shifted toward novel semi-supervised techniques such as the incorporation of word embeddings to represent the context of words and concepts (Tang et al., 2014b). Moreover, it is important to be able to identify sentiment in relation to particular entities, topics or events (aspect-based sentiment).

We have followed a hybrid approach which incorporates traditional lexica, unigrams and bigrams as well as word embeddings using word2vec (Mikolov et al., 2013) to train classifiers for subtasks A and B. For subtask C, sentiment targeted towards a particular topic, we have developed a set of different strategies which use either syntactic dependencies or token-level associations with the topic word in combination with our A classifier to produce sentiment annotations.

2 Phrase-Based Sentiment Analysis (Subtask A) as a Means to an End (subtask C)

Phrase-based sentiment analysis (subtask A) in tweets is a long standing task where the goal is to classify the sentiment of a designated expression within the tweet as either positive, negative or neutral. The state of the art for subtask A achieves high performance usually based on methodologies employing features obtained from either manually or automatically generated lexica (Mohammad et al., 2013; Zhu et al., 2014). However, lexica by definition lack contextual information and are oftain domain dependent. Recent work (Tang et al., 2014a) has successfully used sentiment-specific word embeddings, vector representations of the n-gram context of positive, negative and neutral sentiment in tweets to obtain performance which approaches that of lexicon-based approaches.

Here we employ a combination of lexical features and word embeddings to maximise our performance in task A. We build phrase-based classifiers both with an emphasis on the distinction between positive

Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pages 657–663,
Denver, Colorado, June 4-5, 2015. ©2015 Association for Computational Linguistics

and negative sentiment, which conforms to the distribution of training data in task A, as well as phrase-based classifiers trained on a balanced set of positive, negative and neutral tweets. We use the latter to identify sentiment in the vicinity of topic words in task C, for targeted sentiment assignment. In previous work (Tang et al., 2014a; Tang et al., 2014b) sentiment-specific word embeddings have been used as features for identification of tweet-level sentiment but not phrase-level sentiment. Other work which considered word embeddings for phrase level sentiment (dos Santos, 2014) did not focus on producing sentiment-specific representations and the embeddings learnt were a combination of character and word embeddings, where the relative contribution of the word embeddings is not clear. In this work we present two different strategies for learning phrase level sentiment specific word embeddings.

2.1 Feature Extraction for Task A

Here we provide a detailed description of data preprocessing and feature extraction for phrase-level sentiment. Working on the training set (7,643 tweets), we replaced URLs with "URLINK", converted everything to lower case, removed special characters and tokenised on whitespace, as in (Brody and Diakopoulos, 2011). We decided to keep user mentions, as potentially sentiment-revealing features. We then extracted features both for the *target* (the designated highlighted phrase) and its *context* (the whole tweet):

Ngrams: For a target at the position n in a tweet, we created binary unigram and bigram features of the sequence between $\{n-4, n+4\}$, as suggested by Saif et al. (Mohammad et al., 2013).

Lexicons: We used four different lexica: Bing Liu's lexicon (Hu and Liu, 2004) (about 6,800 polarised terms), NRC's Emotion Lexicon (Mohammad and Turney, 2010) (about 14,000 words annotated based on 10 emotional dimensions), the Sentiment140 Lexicon (62,468 unigrams, 677,968 bigrams and 480,010 non-contiguous pairs) and NRC's Hashtag Sentiment Lexicon (Mohammad et al., 2013) (54,129 unigrams, 316,531 bigrams and 308,808 non-contiguous pairs). We extracted the number of words in the text that appear in every dimension of the Bing Liu and NRC Emotion Lexica. For every lexicon, we extracted features indicating the number of positive unigrams, bigrams and pairs, their maximum sentimental value as indicated by each lexicon, the sum of their sentiment values and the value of the last non-zero (non-neutral) token. All features were extracted both from the tweet as well as the target.

Word Embeddings: We used the tweets collected by (Purver and Battersby, 2012) as training data for sentiment-specific word embeddings. These tweets contain emoticons and hashtags for six different emotions, which we group together to compile positive and negative subsets. To create phrase-level word embeddings, we applied two strategies: *(i)* we searched for positive and negative words (as defined in Bing Liu's lexicon) in the corpus; *(ii)* we performed chi-squared feature selection and extracted the *5,000* most important tokens to be used as our index; for both strategies, we extracted the phrase included in the 2-token-length, two-sided window. The embeddings were learnt by using Gensim (Řehůřek and Sojka, 2010), a Python package that integrates word2vec[1]. In both cases, we created representations of length equal to 100[2]. For each strategy, class and dimension, we used the functions suggested by (Tang et al., 2014b) (average, maximum and minimum), resulting in 2,400 features.

Extra Features: We used several features, potentially indicative of sentiment, a subset of those in (Mohammad et al., 2013). These include: the total number of words of the target phrase, its position within the tweet ("start", "end", or "other"), the average word length of the target/context and the presence of elongated words, URLs and user mentions. We manually labelled various emoticons as positive (strong/weak), negative (strong/weak) and "other" and counted how many times each label appeared in the target and its context.

2.2 Experiments and Results

We experimented with Random Forests and LibSVM with a linear kernel on the training set (4,769 positive, 2,493 negative and 381 neutral tweets) using 10-fold cross-validation and selected LibSVM as the algorithm which achieved the best average F1 score on the positive and negative classes. We then

[1] https://code.google.com/p/word2vec/

[2] The generated, phrase-level Word Embeddings are available at https://zenodo.org/record/14732

used the development set (387 positive, 229 negative and 25 neutral tweets) to fine-tune the value of parameter C, achieving an F1 score of 86.40. The final model was applied on the two test sets provided to us; the “Official 2015 Test” (“OT”) included 3,092 instances and the “Progress Test” (“PT”), including 10,681.

Our results are summarised in Table 1. Our algorithm was ranked fourth in OT and fifth in PT out of 11 competitors, achieving F1 scores of 82.46 and 83.89 respectively. It is clear from the table that lexicon-based features have the most important impact on the results. Interestingly, without ngram features, our results would have been better in both sets; however, there was a 0.8 gain in F1 score with the development set (F1 score 85.60) when these were incorporated in our model. The comparison between all the different pairwise sets of features illustrates that lexica together with word embeddings contribute the most (the results are most affected when they are removed), whereas from the individual feature sets (not presented due to space limitations), lexicon-based features outperform the rest (79.96, 82.18), followed by word embeddings (77.75, 79.92 in OT and PT respectively).

Features Used	**OT**	**PT**
All features	82.46	83.89
– lexica	79.31	80.45
– embeddings	82.01	84.58
– ngrams	**82.72**	**84.63**
– extra	82.37	84.46
– lexica, embeddings	73.11	72.60
– lexica, ngrams	77.70	79.88
– lexica, extra	78.91	80.49
– embeddings, ngrams	79.83	82.66
– embeddings, extra	81.71	84.09
– ngrams, extra	82.64	84.36

Table 1: Average F1 scores of positive/negative classes on the test set with different features.

3 Tweet-Level Sentiment Analysis (Subtask B) Using Multiple Word Embeddings

Our approach to subtask B follows the same logic as for subtask A, feeding a combination of hybrid features (lexical features, n-grams and word embeddings) to an SVM classifier to determine tweet-level polarity. Our approach integrates rich lexicon-based resources and semantic features of tweets, which enables us to achieve an average F1-score of 65.78 on positive tweets and negative tweets in the development dataset. In the final evaluation for subtask B, we got a rank of 27 out of 40 teams for the test dataset and a rank of 24 out of 40 teams for the test progress dataset. The results are discussed in more detail in subsection 3.2.

The features we used are presented below:

3.1 Features

N-grams: We extract unigrams, bigrams and trigrams from the tweets.

Twitter syntax features: These include the number of tokens that are all in uppercase; the numbers of special marks (?, !, #, @); the numbers of positive emoticons (<3, :DD, ;), :D, 8), :-), :), (-:) and the number of negative emoticons (:(, :’(, :/, :-(, :<).

Lexicon-based features: For lexica that only provided the polarities of sentiment bearing words, we used the numbers of matched positive words and negative words in a tweet as features; for lexica that provided sentiment scores for words or ngrams, we included the sum of positive scores of matched words and the sum of negative scores of matched words as two separate features. The lexica we utilised fell into two categories: manually generated sentiment lexica like the AFINN (Nielsen, 2011), MPQA (Wilson et al., 2005), and Bing Liu’s lexica (Liu, 2010); and automatically generated sentiment lexica like the Sentiment140 (Mohammad et al., 2013) and NRC Hashtag Sentiment lexica (Mohammad et al., 2013).

Word embeddings representations features: We learned positive and negative word embeddings separately by training on the HAPPY and NON-HAPPY tweets from Purver & Battersby’s multi-class Twitter emoticon and hashtag corpus (Purver and Battersby, 2012), as with subtask A. The difference with subtask A is that here we used the whole tweet as our input (compared to the two-sided window around a polarised word in subtask A) in order to create tweet-level representations. We set the word embeddings dimension to 100 in order to gain enough semantic information whilst reducing training time.

We also employed the word embeddings encoding sentiment information generated through the unified models in (Tang et al., 2014b). Similar to Tang, we represent each tweet by the min, average, max and sum on each dimension of the word embeddings of all the words in the tweet. In the end, the number of our word embeddings features is $4 \times 100 = 400$. A tweet's representations of word embeddings generated from the HAPPY and non-HAPPY subset of tweets and the embeddings generated by Tang et al. were incorporated into the feature set. Their word embeddings have 50 dimensions, so another $4 \times 50 = 200$ features are added to our feature set.

3.2 Experiments

For our SemEval submission we trained an SVM classifier on 9684 tweets (37.59% positive, 47.36% neutral, 15.05% negative) from the training data set, and used the classifier to classify the 2390 tweets (43.43% positive, 41.30% neutral, 15.27% negative) in the test data set. After the training process, we tested the performance of classifiers with different feature sets (shown in the first column in Table 2) on the development data set (1654 tweets with 34.76% positive, 44.68% neutral, 20.56% negative), and used the average F1 scores of positive and negative tweets as performance measurement. The classifier had the best performance on the development data set, achieving a score of 65.78, compared with 57.32 and 65.47 on the test and test progress datasets. We hypothesize that these differences are caused by differences in the proportions of positive and negative tweets in these datasets.

Experiment	Score
All features	58.53
− positive and negative embeddings	57.32
− n-grams	58.63
− Tang's embeddings	58.83
− Twitter-specific features	58.38
− Manual lexica	57.58
− Automatic lexica	58.39
− All embeddings	56.54

Table 2: The scores obtained on the test set with different features.

In Table 2 we list the average F1 scores of positive and negative tweets in the test data set when removing certain features. The results we submitted were generated by the second classifier. Table 2 demonstrates that representing the tweet with positive and negative word embeddings is the most effective feature (performance is affected the most when we remove these) followed by the manually generated lexicon-based features. This combined with a 2% reduction in F1 score when the embeddings are removed, indicates that the embeddings improve sentiment analysis performance. Contrary to the approach by (Tang et al., 2014b), we didn't integrate the sentiment information in the word embeddings training process, but rather the sentiment-specific nature of the embeddings was reflected in the choice of different training datasets, yielding different word embedding features for positive and negative tweets. To measure the contributions of our word embeddings and Tang's sentiment-specific word embeddings separately in the F1 score, we performed a further test. When we only removed Tang's word embeddings features, the F1 score dropped by 0.15%; when we only removed our word embedding features, the F1 score dropped by 1.21%. This illustrates that for our approach, our word embedding features contribute more. However, it is the combination of the two types of word embeddings that boosts our classifier's performance.

4 Target-Specific Sentiment: Subtask C

Experiment	Score
SUBMISSION	22.79
SUBMISSION-SENTIMENT	29.37
SUBMISSION-RETOKENIZED	27.88
CONLL-PROPAGATION	31.84
BASELINE	46.59
μPARSE	46.87

Table 3: Summary of the performance of our subtask C classifiers.

In subtask C the goal is to identify the sentiment targeted towards a particular topic or entity. This is closely linked to aspect-based sentiment (Pontiki et al., 2014) and is very important for understanding the reasons behind the manifestation of different reactions. We develop several strategies for selecting a topic-relevant portion of a tweet and use it to produce a sentiment annotation. A driving force of our approach has been to use phrase-based sentiment identification from subtask A to annotate the topic-relevant selections.

4.1 Topic Relevance Through Syntactic Relations

A syntactic parser generates possible grammatical relations between words in unstructured text, which are potentially useful for capturing the context around a target topic. We experimented with the Stanford parser (Klein and Manning, 2003) and the recently released TweeboParser (Kong et al., 2014). TweeboParser is explicitly designed to parse tweets – supporting multi-word annotations and multiple roots – but instead of the popular Penn Treebank annotation it uses a simpler annotation scheme and outputs much less dependency type information and was therefore not deemed suitable for our purpose. We used the Stanford parser with a caseless parsing model, expected to work better for short documents. We define the topic-relevant portion of a tweet as the weakly connected components of the dependency graph containing a given topic word.

4.2 Generating Per-Token Annotations

Our four different systems BASELINE, SUBMISSION-SENTIMENT, CONLL-PROPAGATION and μPARSE all use per-token sentiment annotations generated in advance by the linear SVM- and random forest-based classifiers discussed in subtask A, using balanced and imbalanced versions of subtask A's training data. Because the classifier can perform better with additional context, we generated two versions of each annotation set – one token at a time (1-WINDOW), and three at a time (3-WINDOW). 3-WINDOW annotations undergo a further majority pre-processing operation to generate a per-token annotation, since adjacent windows overlap. We found again that the SVM classifier outperformed the random forest classifer, with SUBMISSION-RETOKENIZED and CONLL-PROPAGATION performing best with the balanced version, and μPARSE and BASELINE performing best using the imbalanced training data. In the following we explain each of the above mentioned Task C strategies.

4.3 Using Dependency Relations

CONLL-PROPAGATION builds a dependency graph from a supplied parse, trims some of the relations[3], attaches a 1-WINDOW sentiment to each node using our subtask A classifier, and then propagates those values along variably weighted edges to the target. To help the algorithm propagate successfully, the graph is undirected. We opted to train the edge weights using a simple genetic algorithm. Whilst its performance is modestly better than our submission, the approach is constrained by its inefficiency.

SUBMISSION builds a directed co-dependency graph from the supplied parse, and then attempts to match it against parse trees seen previously, to capture syntactic features that may be relevant to the topic's sentiment. Because subgraph isomorphism is a computationally difficult problem, we use a diffusion kernel (as in (Fouss et al., 2006)) to normalise the adjacency matrix for SVM classification. We also add unigrams within the same window used for BASELINE as an additional feature. SUBMISSION-RETOKENIZED updates the result and replaces whitespace tokenization with that used by (Gimpel et al., 2011), more aggressively trims the adjacency matrix, and improves the pre-processing pipeline, improving performance a little. SUBMISSION-SENTIMENT changes the structure of the dependency graph by connecting tokens to their 1-WINDOW sentiment derived from task A, improving performance further still.

4.4 Classification Without Dependency Relations

The simplest classification method (BASELINE) identifies the topic and then only considers those tokens around it. Despite being rudimentary, we found BASELINE difficult to beat when teamed with the sentiment analyser developed for part A, producing an F1-score of 46.59 with a window of 8 tokens. BASELINE is also useful because it doesn't require the use of the training data for task C, leaving it free for validation.

μPARSE is an approach offering a compromise between potentially noisy dependency parsing and the

[3] We select 9 dependency relations – 'amod', 'nsubj', 'advmod', 'dobj', 'xcomp', 'ccomp', 'rcmod', 'cop' and 'acomp' which feasibly impact sentiment (Li et al., 2011).

model-free baseline. It feeds a buffer of word2vec-derived word representations into *min-max-average* feature map (similar to (Tang et al., 2014a)), which is then classified with a linear SVM to decide whether to segment the tweet at the position of an incoming token, or to add the current token to the existing segment. The aim is to extract the neighbourhood around the root words that would have been identified by a perfect syntactic parser, making it conceptually similar to chunking. μPARSE then seeks out a cluster containing the target concept and then uses 1-WINDOW or 3-WINDOW to obtain a consensus annotation. When trained and evaluated on TweeboParser's dataset, it incorrectly groups root words together at a rate of 16%, but this is sufficient to slightly outperform BASELINE.

4.5 Discussion

We found it surprising that our task C submission did not need to be very complex, since we could determine phrase-level sentiment accurately. Whilst we decided not to submit BASELINE as our official entry – owing to uncertainty about the best subtask A parameters and its lack of technical sophistication – our results in Table 3 clearly demonstrate that we should have done so. Syntactic information does seem effective on its own in combination with phrase-level sentiment data, but its real utility might be to guide a more advanced approach that detects syntactically complex structures, and cedes the rest to BASELINE.

5 Conclusions

We have presented our system's components for phrase-, tweet- and topic-based sentiment classification. While lexica remain a critical aspect of our system, we have found that word embeddings are highly important and have great potential for future research in this domain. For both subtasks A and B we generated sentiment-specific word embeddings which yield a performance comparable to that of our lexicon-based approach and further enhance performance. Furthermore, we have found that syntactic features can be useful for topic-based sentiment classification, achieving good results when combined with phrase-based sentiment labels. However, our findings also indicate that simpler approaches can perform better (perhaps due to the need for improvements in dependency parsing for Twitter), and further investigation will be required to determine how to exploit the relationship between topic-specific and phrase-level sentiment.

Acknowledgements

This work was partially funded by the Engineering and Physical Sciences Research Council (grant EP/L016400/1) through the University of Warwick's Centre for Doctoral Training in Urban Science and Progress.

References

Samuel Brody and Nicholas Diakopoulos. 2011. Cooooooooooooooollllllllllllll!!!!!!!!!!!!!!: using word lengthening to detect sentiment in microblogs. In *Proceedings of the conference on empirical methods in natural language processing*, pages 562–570. Association for Computational Linguistics.

Cicero dos Santos. 2014. Think positive: Towards Twitter sentiment analysis from scratch. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 647–651, Dublin, Ireland, August.

Francois Fouss, Luh Yen, Alain Pirotte, and Marco Saerens. 2006. An experimental investigation of graph kernels on a collaborative recommendation task. In *Data Mining, 2006. ICDM'06. Sixth International Conference on*, pages 863–868. IEEE.

Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A. Smith. 2011. Part-of-speech tagging for Twitter: Annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, HLT '11, pages 42–47, Stroudsburg, PA, USA.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 423–430, Stroudsburg, PA, USA.

Lingpeng Kong, Nathan Schneider, Swabha Swayamdipta, Archna Bhatia, Chris Dyer, and

Noah A Smith. 2014. A dependency parser for tweets. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, to appear*.

Peifeng Li, Qiaoming Zhu, and Wei Zhang. 2011. A dependency tree based approach for sentence-level sentiment classification. In *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2011 12th ACIS International Conference on*, pages 166–171. IEEE.

Bing Liu. 2010. Sentiment analysis: a multifaceted problem. *IEEE Intelligent Systems*, 25(3):76–80.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

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 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pages 26–34. Association for Computational Linguistics.

Saif Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. 2013. Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. In *Proceedings of the seventh international workshop on Semantic Evaluation Exercises (SemEval-2013)*, Atlanta, Georgia, USA, June.

Finn Årup Nielsen. 2011. A new anew: Evaluation of a word list for sentiment analysis in microblogs. *arXiv preprint*.

Maria Pontiki, Haris Papageorgiou, Dimitrios Galanis, Ion Androutsopoulos, John Pavlopoulos, and Suresh Manandhar. 2014. Semeval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35.

Matthew Purver and Stuart Battersby. 2012. Experimenting with distant supervision for emotion classification. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 482–491. Association for Computational Linguistics.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May.

Duyu Tang, Furu Wei, Bing Qin, Ting Liu, and Ming Zhou. 2014a. Coooolll: A deep learning system for Twitter sentiment classification. *SemEval 2014*, page 208.

Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. 2014b. Learning sentiment-specific word embedding for Twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 1555–1565.

Mike Thelwall and Kevan Buckley. 2013. Topic-based sentiment analysis for the social web: The role of mood and issue-related words. *Journal of the American Society for Information Science and Technology*, 64(8):1608–1617.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 347–354, Stroudsburg, PA, USA.

Xiaodan Zhu, Svetlana Kiritchenko, and Saif Mohammad. 2014. Nrc-canada-2014: Recent improvements in the sentiment analysis of tweets. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 443–447, Dublin, Ireland, August.