IBC-C: A Dataset for Armed Conflict Event Analysis

Andrej Žukov-Gregorič and **Bartal Veyhe** and **Zhiyuan Luo**
Department of Computer Science
Royal Holloway, University of London
Egham TW20 0EX
{andrej.zukovgregoric.2010, bartal.veyhe.2014}@live.rhul.ac.uk
zhiyuan@cs.rhul.ac.uk

Abstract

We describe the Iraq Body Count Corpus (IBC-C) dataset, the first substantial armed conflict-related dataset which can be used for conflict analysis. IBC-C provides a ground-truth dataset for conflict specific named entity recognition, slot filling, and event de-duplication. IBC-C is constructed using data collected by the Iraq Body Count project which has been recording casualties resulting from the ongoing war in Iraq since 2003. We describe the dataset's creation, how it can be used for the above three tasks and provide initial baseline results for the first task (named entity recognition) using Hidden Markov Models, Conditional Random Fields, and Recursive Neural Networks.

1 Introduction

Many reports about armed conflict related incidents are published every day. However, these reports on the deaths and injuries of civilians and combatants often get forgotten or go unnoticed for long periods of time. Automatically extracting casualty counts from such reports would help better track ongoing conflicts and understand past ones.

Casualty counting, a subfield of conflict analysis, can be split into two distinct approaches (Seybolt et al., 2013). A *statistical* approach which uses sampling methods to infer total casualty counts (Burnham et al., 2006; Price et al., 2014) and a *direct recording* approach which identifies and counts individual casualties.

One popular direct recording approach is to identify incidents from textual reports and extract casualty information from them. This can either be done by hand or automatically. The Iraq Body Count project (IBC) has been directly recording casualties since 2003 for the ongoing conflict in Iraq (IBC, 2016; Hicks et al., 2011). IBC staff collect reports, link them to unique incidents, extract casualty information, and save the information on a per incident basis as can be seen in Table 2.

Direct recording by hand is a slow process and notable efforts to do so have tended to lag behind the present. Information extraction systems capable of automating this process must explicitly or implicitly successfully solve three tasks: (1) find and extract casualty information in reports (2) detect events mentioned in reports (3) deduplicate detected events into unique events which we call *incidents*. The three tasks correspond to named entity recognition, slot filling, and de-duplication.

In this work we introduce the report based IBC-C dataset.[1] Each report can contain one or more sections; each section, one or more sentences; each sentence, one or more words. Each word is tagged with one of nine entity tags in the inside-outside-beginning (IOB) style. A visual representation of the dataset can be seen in Figure 1 and its statistics in Table 1.

To the best of our knowledge apart from the significantly smaller MUC-3 and MUC-4 datasets (which aren't casualty-specific) there are no other publicly available datasets made specifically for tasks (1), (2) or (3). The IBC-C dataset can be used to train supervised models for all three tasks.

We provide baseline results for task (1) which we posit as a sequence-classification problem and solve using an HMM, a CRF, and an RNN.

Since the 1990s the conflict analysis and NLP/IE communities have diverged. With the IBC-C dataset we hope to bring the two communities closer again.

[1] More information about the IBC-C dataset can be found on: http://andrejzg.github.io/ibcc/

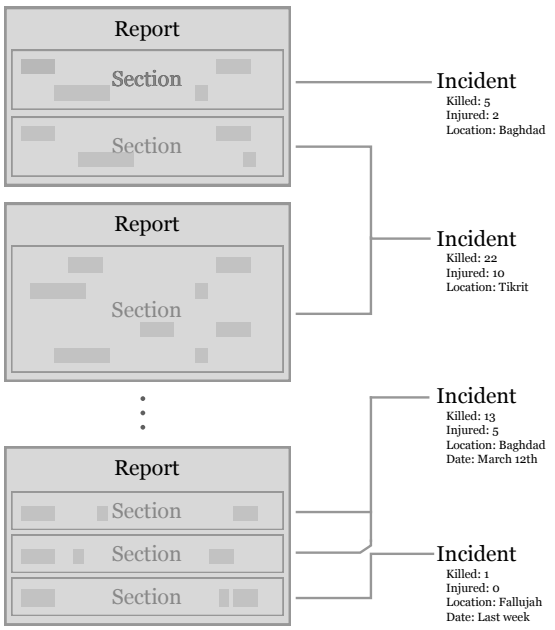

Figure 1: The IBC-C dataset visualised. A report is split into one or more non overlapping sections. A section is comprised of sentences which are comprised of words. Each section is linked to exactly one incident which in turn can be linked to one or more sections.

Element	Count
incidents	9,184
sections	18,379
reports	16,405
sentences	35,295
words	857,465
KNUM	13,597
INUM	6,689
KSUB	14,395
ISUB	1,036
KOTHER	1,192
IOTHER	495
LOCATION	25,251
DATE	4,765
WEAPON	35,617

Table 1: Dataset statistics. Fully capitalised words indicate named entity tags.

2 Related Work

Extracting information from conflict related reports has been a topic of interest at various times for both the conflict analysis, information extraction, and natural language processing communities.

The 1990s saw a series of message understanding conferences (MUCs) of which MUC-3 and MUC-4 are closely related to our work and contain reports of terrorist incidents in Central and South America. MUC data is most often used for slot filling and although MUC-3 and MUC-4 contain more slots than IBC-C they are at the same time much smaller (MUC4 contains 1,700 reports) and cannot be used for incident de-duplication.

Although various ACE, CoNNL, and TAC-KBP tasks contain within them conflict-related reports, none of them are specific to conflict and haven't been studied for conflict-related information extraction specifically.

Studies more directly related to our dataset include work by Tanev and Piskorski (Tanev et al., 2008) who use pattern matching to count casualties. They report a 93% accuracy on counting the wounded. However, they have access to only 29 unique conflict events. Other non-casualty conflict-related work in the domain also suffers from a lack of data, for example, (King and Lowe, 2003) only deal with 711 reports.

Despite work in the NLP and IE communities, the conflict analysis community is still reliant on datasets created by hand. These include IBC (IBC, 2016), ACLED (Raleigh et al., 2010), EDACS (Chojnacki et al., 2012), UCDP (Gleditsch et al., 2002), and GTD (GTD, 2015).

To the best of our knowledge there are no efforts to fully automate casualty counting. However, efforts using NLP/IE tools to automate incident detection do exist but their ability to de-deduplicate incidents has been called into question (Weller and McCubbins, 2014).

Three notable such efforts originating in the conflict analysis community are GDELT (Leetaru and Schrodt, 2013), ICEWS (Obrien, 2010), and OEDA (Schrodt, 2016). All three use pattern matching software such as TABARI (Schrodt, 2001) and to categorise reports using the CAMEO coding scheme (Schrodt et al., 2008).

3 Creating the IBC-C Dataset

3.1 Preprocessing

The Iraq Body Count project (IBC) has been recording conflict-related incidents from the Iraq war since 2003. An incident is a unique event related to war or other forms of violence which led to the death or injury of people. An example can be seen in Table 2.

The recording of incidents by the IBC works as follows: IBC staff first collect relevant *reports* before highlighting *sections* of them which they deem relevant to individual incidents. Parts of

Incident ID	Start date	End date
d3473	22 Mar 2003	22 Mar 2003
Min killed	**Max killed**	**Min injured**
2	2	8
Max injured	**Location**	**Cause of death**
9	Khurmal	Suicide car bomb
Sources	**Town**	**Province**
BBC 23 Mar DPA 23 Mar	Khurmal	Sulaymaniyah
Alt. province	**District**	**Alt district**
/	Halabja	/
Killed Subjects Person 1, Person 2, ...		
Injured Subjects Person 3, Person 4, ...		
Report Sections BBC: "On Saturday **Person 1** died in **Khurmal** ..." DPA: "**2** people died yesterday afternoon..."		

Table 2: An example of an incident hand coded by IBC staff. Min and max values represent the minimum and maximum figures quoted in report sections linked to the incident.

the report outside the highlighted sections are discarded. Sections can be seen in Figure 1. Because of the way IBC staff highlight sections there are no overlapping sections in the IBC-C dataset. Events are then recognised from the highlighted sections and de-duplicated into incidents. A final description of the incident (e.g. death and injury counts, location and date) is agreed upon after multiple rounds of human checking.

In the preprocessing step we gathered all incidents which occurred between March 20th, 2003 and December 31st, 2013. We removed spurious incidents (e.g. where the minimum number killed is larger than the maximum number killed) and cleaned the section text by removing all formatting and changing all written-out numbers into their numeric form (e.g. 'three' to 3).

3.2 Annotation

Using the information extracted by the IBC (see Table 2) we annotated each section word with one of ten tags: *KNUM* and *INUM* for numbers representing the number killed and injured respectively; *KSUB* and *ISUB* for named individuals were killed or injured; *KOTHER* and *IOTHER* for unnamed people who were killed or injured (for example "The doctor was injured yesterday."); *LOCATION* for the location in which an incident occurred; *WEAPON* for any weapons used in an attack; *DATE* for words which identify when the incident happened; and, *O* for all other words.

Our data generation process can be thought of as a form of *distant supervision* (Mintz et al., 2009) where we use agreed upon knowledge about an incident to label words contained within its sections instead of having hand-labeled individual words. This inevitably introduces errors which we try to mitigate using a filtration step where we remove ambiguous data.

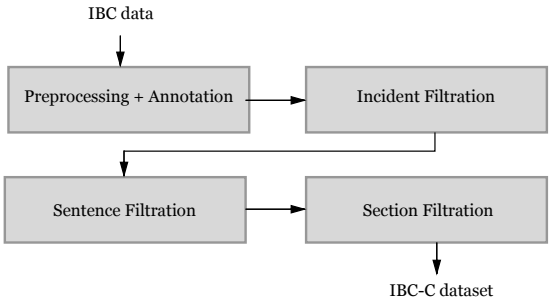

Figure 2: A visualisation of the different steps taken to create the dataset.

3.3 Filtration

Simply annotating words based on the information in Table 2 can lead to wrong annotations. For example, if two people were recorded as having died in an incident, then, if another number two appears in the same sentence, this might lead to a wrong annotation. The sentence, "2 civilians were killed after 2 rockets hit the compound" could lead to the second '2' being annotated as a KNUM. The actual *cardinality* of a number makes little difference to a sequence classifier compared to the difference a *misannotated* number would make. To minimise such misannotations we remove sentences and reports which do not pass all filtration criteria. Our filtration criteria consist of boolean functions over sentences, sections and incidents which return false if a test isn't passed.

The goal of filtration is to remove as much ambiguously labelled data as possible without biasing against any particular set of linguistic forms. There is thus a tradeoff which must be struck between linguistic richness and the quality of annotation.

In our case we found that simple combinations of pattern matching and semantic functions, as in 3, worked well. No syntactic functions were used.

3.3.1 Incident Filtration

Incidents are filtered using a single criterion: if the minimum number of people killed or injured does not equal the maximum number of people killed or injured, respectively, (Table 2) then the incident is removed. We do this so as to minimise any ambiguity in our named entity tagging (the only task for which we provide baseline results). This

hasKNUM	isKillSentence	hasOneTaggedAsKNUM	hasNumber	otherKNUMsInSection	toConsider	#
+	+	+	+	+	-	2,445
+	+	+	+	-	+	7,526
+	+	-	+	+	-	14,624
+	+	-	+	-	+	30,204
+	-	+	+	+	-	2,119
+	-	+	+	-	-	1,498
+	-	-	+	+	-	4,282
+	-	-	+	-	-	4,648
-	+	-	+	+	+	2,757
-	+	-	+	-	-	67,402
-	+	-	-	+	+	3,360
-	+	-	-	-	+	43,006
-	-	-	+	+	+	7,573
-	-	-	+	-	+	47,736
-	-	-	-	+	+	19,749
-	-	-	-	-	+	125,010

Table 3: Filtration criteria. An example of a set of boolean functions (columns one through five) applied to sentences to filter out ambiguous KNUM annotations. Sentences which we wish to allow are identified by a '+' in the *toConsider* column. Sentence counts are given in the last column. Only rows with non-zero counts are shown. Shaded rows indicate sentences which are ambiguous are shaded and identified by a '-'. We show only the KNUM table due to lack of space.

has the adverse effect of removing any incidents where reports mention different casualty counts. To compile a dataset which disregards this criterion, or considers a permissible window of casualties, a parameter in our dataset generating program may be changed.

3.3.2 Sentence Filtration

Filtering sentences is by far the hardest step. It is here where we must be careful to not bias against any linguistic forms. A separate set of boolean functions are applied to each sentence for the KNUM and INUM entity tags. An example for the KNUM tag can be seen in Table 3. Every sentence passes through four boolean functions (the first four columns) and is then labeled as either having passed or failed the test (fifth column). The fifth column was decided upon by us in advance.

In the case of Table 3: *hasKNUM* indicates whether the sentence contains a word tagged as KNUM; *isKillSentence* indicates whether any of its words are connected to death or killing (by matching them against a list of predefined words); *hasOneTaggedAsKNUM* indicates whether the number '1' is tagged as a KNUM (remember that we convert written out numbers such as 'three' to '3' and that 'one', and thus '1', can also be a pronoun); *hasNumber* indicates whether a sentence has a number; and, *otherKNUMsInSection* indicates whether there are other words tagged as KNUM in the section.

3.3.3 Report Filtration

Report filtering is simple and again done using only one rule. If any sentence a report contains fails to pass a single sentence-level test, then the whole report is removed.

3.4 Tasks

3.4.1 Named Entity Recognition

Each word in the IBC-C dataset is tagged with one of nine (excluding *O*) entity tags as can be seen in Table 1 which can be thought of as subsets of more common named entity tags such as person or location. The dataset can be used to train a supervised NER model for conflict-specific named entity tags. This is important for relationship extraction which relies on good named entity tags.

3.4.2 Slot Filling and Relationship Extraction

Each IBC-C event can be thought of as a 9-slot *event* template where each slot is named after an entity tag. The important thing to keep in mind is that a report may contain more than one section so just correctly recognising the entities isn't enough to solve the slot filling task. Instead, if a report mentions two events then two separate templates must be created and their slots filled.

A common sub-problem of slot filling is relationship extraction. Because we know which incident every section refers to, generating ground-truth relationships is trivial because we may be sure that an entity which appears in one of the sections is related to every other entity in that same section. For example, finding a KSUB and a LOCATION means that we can build a *killed_in(KSUB, LOCATION)* relationship.

3.4.3 Event De-duplication

Since the IBC-C dataset preserves the links between sections and incidents it may be used as

	HMM			CRF 13-window			RNN 13-window		
Tag	Precision	Recall	**F1**	Precision	Recall	**F1**	Precision	Recall	**F1**
KNUM	0.63	0.86	**0.73**	0.91	0.94	**0.92**	0.90	0.85	**0.88**
INUM	0.50	0.39	**0.44**	0.95	0.93	**0.94**	0.87	0.91	**0.89**
KSUB	0.73	0.68	**0.70**	0.82	0.76	**0.79**	0.86	0.53	**0.66**
ISUB	0.00	0.00	**0.00**	0.89	0.24	**0.38**	0.80	0.06	**0.12**
KOTHER	0.39	0.19	**0.25**	0.83	0.54	**0.66**	0.41	0.36	**0.38**
IOTHER	0.00	0.00	**0.00**	0.80	0.61	**0.69**	0.55	0.50	**0.52**
LOCATION	0.75	0.70	**0.73**	0.85	0.77	**0.80**	0.86	0.70	**0.77**
DATE	0.75	0.64	**0.69**	0.75	0.64	**0.69**	0.41	0.30	**0.35**
WEAPON	0.98	0.89	**0.93**	0.98	0.90	**0.94**	0.97	0.87	**0.92**
Overall	0.57	0.53	**0.55**	0.88	0.73	**0.78**	0.74	0.57	**0.61**

Table 4: Results for various models

a ground-truth training set for training event de-duplication models.

4 Experiments

Baseline results were computed for the named entity recognition task using an 80:20 tag split across sentences (we ignore report or section boundaries). We compare three different sequence-classification models as seen in Table 4: a Hidden Markov Model (Zhou and Su, 2002), a Conditional Random Field (McCallum and Li, 2003), and a Elman-style Recursive Neural Network similar to the one used in (Mesnil et al., 2013).

For the HMM we use bigram features in combination with the current word and the current base named entity features[2]. We trained the HMM in CRF form using LBFGS.

For the CRF we find that using bigram features and a 13-word window, across words and base named entities, gives us the best result. We train the CRF using LBFGS. All CRF training, including the HMM, was done using CRFSuite (Okazaki, 2007).

For the Elman-style recurrent network we use randomly initialised 100 dimensional word vectors as input, the network has 100 hidden units, and we use a 13-word context window again. The RNN was implemented using Theano (Bastien et al., 2012). We train the RNN using stochastic gradient descent on a single GPU.

4.1 Evaluation

The first thing which strikes us is how low the ISUB scores are. The CRF returns a recall score of 0.24. At the same time, the precision is relatively high at 0.89. Low recall indicates a lot of false negative classifications - i.e. there were many injured people who were mistakingly tagged as uninjured. A high precision rate means a low false positive rate - i.e. most uninjured people were correctly tagged as uninjured. In short, the classifier was too generous with tagging people as having been injured. Looking at the dataset we realise that in contrast to KSUBS, words which we associate with injury such as "wounded" or "injured" are often very far away from an ISUB. Increasing the window size with the CRF didn't help (such large features are often never expressed during the test phase).

Low recall scores across multiple tags indicate that long-distance dependencies determine a word's classification. K/INUM recall is exceptionally high because K/INUMs are usually surrounded by words such as "killed". We were surprised to see the RNN perform relatively poorly and expected it to be able to factor in long-distance dependencies. We believe this has more to do with our hyper-parameter settings than deficiencies in the actual model.

5 Conclusion

We present IBC-C, a new dataset for armed conflict analysis which can be used for entity recognition, slot filling, and incident de-duplication.

6 Acknowledgements

We would like to thank members of the IBC, especially Hamit Dardagan for his help with procuring and helping us understand the data collected by the IBC. We would also like to thank Gregory Chockler, Mike Spagat, and Andrew Evans for their insightful discussions and suggestions. This work was partially supported by EPSRC grant EP/K033344/1 ('Mining the Network Behaviour of Bots').

[2]Base named entities such as PERSON and LOCATION were found using Stanford's named entity recogniser (Finkel et al., 2005).

References

Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, James Bergstra, Ian Goodfellow, Arnaud Bergeron, Nicolas Bouchard, David Warde-Farley, and Yoshua Bengio. 2012. Theano: new features and speed improvements. *arXiv preprint arXiv:1211.5590*.

Gilbert Burnham, Riyadh Lafta, Shannon Doocy, and Les Roberts. 2006. Mortality after the 2003 invasion of iraq: a cross-sectional cluster sample survey. *The Lancet*, 368(9545):1421–1428.

Sven Chojnacki, Christian Ickler, Michael Spies, and John Wiesel. 2012. Event data on armed conflict and security: New perspectives, old challenges, and some solutions. *International Interactions*, 38(4):382–401.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370. Association for Computational Linguistics.

Nils Petter Gleditsch, Peter Wallensteen, Mikael Eriksson, Margareta Sollenberg, and Håvard Strand. 2002. Armed conflict 1946-2001: A new dataset. *Journal of peace research*, 39(5):615–637.

GTD. 2015. Global terrorism database. `http://www.start.umd.edu/gtd`. (Accessed on 02/23/2016).

Madelyn Hsiao-Rei Hicks, Hamit Dardagan, Gabriela Guerrero Serdán, Peter M Bagnall, John A Sloboda, and Michael Spagat. 2011. Violent deaths of iraqi civilians, 2003–2008: analysis by perpetrator, weapon, time, and location. *PLoS Med*, 8(2):e1000415.

IBC. 2016. Iraq body count. `https://www.iraqbodycount.org/database/`. (Accessed on 02/23/2016).

Gary King and Will Lowe. 2003. An automated information extraction tool for international conflict data with performance as good as human coders: A rare events evaluation design. *International Organization*, 57(03):617–642.

Kalev Leetaru and Philip A Schrodt. 2013. Gdelt: Global data on events, location, and tone, 1979–2012. In *ISA Annual Convention*, volume 2. Citeseer.

Andrew McCallum and Wei Li. 2003. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 188–191. Association for Computational Linguistics.

Grégoire Mesnil, Xiaodong He, Li Deng, and Yoshua Bengio. 2013. Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding. In *INTERSPEECH*, pages 3771–3775.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. 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 2-Volume 2*, pages 1003–1011. Association for Computational Linguistics.

Naoaki Okazaki. 2007. Crfsuite: a fast implementation of conditional random fields (crfs). (Accessed on 02/23/2016).

Sean P Obrien. 2010. Crisis early warning and decision support: Contemporary approaches and thoughts on future research. *International Studies Review*, 12(1):87–104.

Megan Price, Jeff Klingner, Anas Qtiesh, and Patrick Ball. 2014. Updated statistical analysis of documentation of killings in the syrian arab republic. *Human Rights Data Analysis Group, Geneva*.

Clionadh Raleigh, Andrew Linke, Håvard Hegre, and Joakim Karlsen. 2010. Introducing acled: An armed conflict location and event dataset special data feature. *Journal of peace research*, 47(5):651–660.

Philip A Schrodt, Omür Yilmaz, Deborah J Gerner, and Dennis Hermreck. 2008. The cameo (conflict and mediation event observations) actor coding framework. In *2008 Annual Meeting of the International Studies Association*.

Philip A Schrodt. 2001. Automated coding of international event data using sparse parsing techniques. In *annual meeting of the International Studies Association, Chicago*.

Philip A Schrodt. 2016. Open event data alliance (oeda). (Accessed on 02/23/2016).

Taylor B Seybolt, Jay D Aronson, and Baruch Fischhoff. 2013. *Counting Civilian Casualties: An Introduction to Recording and Estimating Nonmilitary Deaths in Conflict*. Oxford University Press.

Hristo Tanev, Jakub Piskorski, and Martin Atkinson. 2008. Real-time news event extraction for global crisis monitoring. In *Natural Language and Information Systems*, pages 207–218. Springer.

Nicholas Weller and Kenneth McCubbins. 2014. Open event data alliance (oeda)raining on the parade: Some cautions regarding the global database of events, language and tone dataset. (Accessed on 05/18/2016).

GuoDong Zhou and Jian Su. 2002. Named entity recognition using an hmm-based chunk tagger. In *proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 473–480. Association for Computational Linguistics.