Arab Spring: from newspaper data to forecasting

Kenneth Joseph[ij], Kathleen M. Carley[ij], David Filonuk[ij],
Geoffrey P. Morgan[ij], Jürgen Pfeffer[ij]

[i] Computation, Organizations and Society Program
School of Computer Science
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA, USA 15213

[j] kjoseph, kathleen.carley, dfilonuk, gmorgan, jpfeffer @ cs.cmu.edu

Corresponding author:
Kenneth Joseph
Phone: (716) 983-4115
Fax: (412) 268-1744

Abstract

Agent-based simulation models are an important methodology for explaining social behavior and forecasting social change. However, a major drawback to using such models is that they are difficult to instantiate for specific cases and so are rarely re-used. We describe a text-mining network analytic approach for rapidly instantiating a model for predicting the tendency toward revolution and violence based on social and cultural characteristics of a large collection of actors. We illustrate our approach using an agent-based dynamic-network framework, Construct, and newspaper data for the sixteen countries associated with the Arab Spring. We assess the overall accuracy of the base model across independent runs for twenty different months during the Arab Spring, observing that although predictions led to several false positives, the model is able to predict revolution before it occurs in three of the four nations in which the government was successfully overthrown.

Keywords

Agent-based modeling, social simulation, model instantiation, Arab Spring

1. Introduction

Computer simulation models, in particular agent-based models and system dynamic models, are a critical methodology for supporting human reasoning about complex socio-cultural systems (Epstein and Axtell 1996; Gilbert 2007). A "feature" of these technologies is that as the models become more realistic, they become more capable both diagnostically and predictively. Creating such a realistic model generally requires instantiating it using a wide variety of data for the case in question. Such data often needs to be extracted from a wide range of sources, put into the same time frame, fused and otherwise massaged before it can be used to instantiate the model. Thus, the process of using data to instantiate the model often takes as long or longer than it does to build and debug the model. Due to the huge person-time cost of instantiating a model, shortcuts are often built into the simulation process to make special use of special data. The result is methodologies, including toolkits for instantiation and entire simulation engines, which are basically usable only for the context for which they have been instantiated.

To increase the usability of computer simulation technology to study socio-cultural phenomena, we see two methodological pieces that are necessary. First, a validated, stable simulation engine that provides general and basic mechanisms of human social behavior. On top of this framework, models for specific questions (e.g. state revolution) can then be rapidly pieced together by domain experts without worry of the validity of the underlying simulation engine. A stable underlying simulation engine also provides a stable interface for input that is not amenable to unique processes or data hard-coded into the simulation engine itself. The second methodological tool is a means for automatically, or at least more automatically, instantiating simulation models with empirically grounded data. Reducing the challenge of building a simulation, providing a uniform mechanism for data entry and the automatic instantiation of models make for simulations that are easier to reuse and results that are easier to validate. See Carley et al. (2012) for a discussion of validation techniques.

The focus of the present work is to provide an overview of recent developments along these two methodological fronts. We give an approach that reduces the instantiation challenge for computer simulation models by extracting data for the model from online information sources using automated and semi-automated processes. This data is then fed into the dynamic-network modeling framework Construct (Carley 1991; Carley 1990), and in particular a *multi-level* extension of Construct. Construct accepts data in a uniform format, specifically in the form of a series of networks, which allows the framework to be used independently of the data at hand. Finally, on top of Construct, we build a model specific to the context at hand.

Our work expands on previous efforts (Lanham, Morgan, and Carley 2014) by introducing technology that scales better to large numbers of agents and that is

more rigorously automated. These efforts define the beginning of a new set of methodologies that have the potential to make simulation models of large social systems easier to build, reuse, and validate. While our approach does not generalize to all contexts, the methods described in the present work lay important ground-work for domain and data-agnostic simulation models.

Two advances have made our approach possible. First, advances in text mining that support the extraction of multi-mode, multi-link networks, i.e., meta-networks, from texts have made it possible to automatically extract and make sense of relevant data. Second, the evolution of agent-based models into agent-based dynamic-network models have changed the basic representation of the data needed for instantiation from rules to networks, specifically meta-networks (Carley, 2002). Thus, our work is particularly valuable for agent-based dynamic-network models where the actors of interest are discussed in documents that can be text-mined and those that fit into the social mechanisms described by Construct.

Herein we illustrate our advances by applying them to the Arab Spring, a series of revolutionary activities, some of which were violent, that began in December 2010, and in some cases are still ongoing. We pulled over 400,000 news articles from Lexis-Nexis from July 2010 to February of 2012 and used unsupervised and semi-supervised machine learning techniques to draw out the meta-networks necessary for instantiation of an agent-based dynamic network model. The focus of the model is on the dispersion of two beliefs – one centered on the need for revolution and one centered on the need for violence. The model reflects the idea that there are multiple constituencies, linked in a network, each with a set of knowledge. Some of this knowledge leads agents to view revolution and/or violence positively or negatively. This model is run using Construct in tandem with the empirical data drawn from the newspaper text we study.

While much work has considered the Arab Spring, most has been in the context of how new media affected the revolution (e.g. Hamdy and Gomaa 2012; Papacharissi and de Fatima Oliveira 2012) or in developing a better understanding of specific long-term factors that led to revolution (e.g. education, Campante and Chor 2012). In contrast, we focus on uncovering processes in the short term that may have led to violence and revolution, but also those that led certain nations to retain stability and peace. These efforts are more in line with recent work by Pfeffer and Carley (2012), who took a portion of the data used in the present article and explored the dynamics of different terms during the Arab Spring.

Our model generates predictions for each of the twenty months of data independently on a set of sixteen countries that have been identified as having the potential to be swept up in the frenzy of the Arab Spring. This generates, in essence, twenty re-uses of the basic state revolution model, showing the ability of our methodology to be reused with different data sets. Results are analyzed on a per country basis and contrasted with what actually happened using an outside data source that gives the time at which national governments were (or were not)

overthrown. The model correctly predicts three out of four successful revolutions before they occurred, predicting the fourth successful revolution two months after it occurred. Importantly, however, the model also predicts revolution in six countries where a successful revolution did not occur. While significant revolutionary activity did exist in two of these countries, our results thus suggest that future tuning of the model is necessary and that results from such models should be evaluated conservatively.

The paper continues as follows: in Section 2, we provide a methodological overview to give the reader a broad overview of the technical approach taken. Section 3 details the simulation framework, Construct, used in the present work, and Section 4 details the state revolution simulation model we build on top of Construct. Section 5 details how we move from newspaper data to the data necessary to instantiate the model. Section 6 provides results of the simulations[1] and Section 7 concludes with limitations and prospects for future work.

2. Methodological Overview

FIGURE 1 ABOUT HERE

FIGURE 2 ABOUT HERE

Figure 1 illustrates the workflow used in this article. We begin by pulling newspaper articles from major English-based world news drawn from LexisNexis Academic. LexisNexis uses a proprietary algorithm to associate each article with a set of *index terms*. We query the LexisNexis database for any articles indexed by one of the eighteen country names shown in the legend of the map in Figure 2 from the period of July 2010 to February 2012. In total, approximately 400,000 articles were collected. Note that while the original data collection used these eighteen keywords, only sixteen countries were used for analysis. This is because we were not able to obtain enough data on two nations, Western Sahara and Qatar, to merit further investigation.

After obtaining the data, we split it up by month[2] and generate a meta-network (detailed below) that serves as the input for our simulation model. Much work has been done in the area of drawing social structure from text. Recent efforts have made a distinct push towards semi-supervised (e.g. Diesner and Carley 2008) and unsupervised (e.g. Eisenstein et al. 2010) learning approaches to extracting social

---

[1] Code used for this article, as well as a sample of the data, can be found at https://github.com/kennyjoseph/arabspring

[2] We will hereafter refer to a *set of articles* as the collection of all articles within a single month.

information from text to combat the volume of information that can readily be extracted from the web today. In the present work, our focus is on providing a suite of both semi-supervised and unsupervised techniques that draw out social networks, or more specifically, meta-networks, from text. This approach is grounded in recent efforts showing the practicality of the meta-network concept when pulling social structure from textual data (Diesner and Carley 2008; Diesner, Frantz, and Carley 2005; Lanham, Morgan, and Carley 2014; Martin, Pfeffer, and Carley 2013) though the methods are distinctly unique.

After drawing out our meta-networks from the text, they are passed to Construct as-is along with the simulation model we build specifically to analyze the Arab Spring. Thus, the meta-network we generate from the text is used to *instantiate* the simulation model. Note that each month, because the simulations are based on stochastic processes, it was necessary to complete replications of each model. Each of the instantiated models was simulated eight times using slightly different parameterizations of the model, as prior study on the number of replications needed to generate robust results using Construct showed that anything more than five runs were sufficient for result stability (Lee and Carley, n.d.). Thus, this number of replications was deemed sufficient to generate a robust ensemble estimate of the average time to revolution or outbreak of significant violence.

Finally, we use model output to determine the point, if any, at which our approach predicts a revolution in a particular nation is likely. While this outcome could be compared to a variety of real-world data sources, we choose to compare decisions made from model output to real-world data on nations involved in the Arab Spring where the government in power was successfully overthrown.

In the following three sections, we detail our simulation framework, Construct, the specific model instantiated here and how we extract the meta-network necessary to instantiate the model. We provide this discussion in the reverse order that these processes are introduced in Figure 1, as an explanation of the data needed by the model allows a better understanding of the techniques used to pull meta-networks from the newspaper articles.

3. Construct

Humans tend to hold social relationships with those that they are similar to (Lazarsfeld and Merton 1954; McPherson, Lovin, and Cook 2001;Kossinets and Watts 2009). These relationships are bound together by persistent interaction (Licoppe 2004). When humans interact with similar others, they share information and beliefs. This sharing, when moved beyond the context of two individuals in a social relationship into a dynamic social network, creates cascades of information and beliefs that *diffuse* throughout the network (Rogers 2003).

A plethora of methods have been put forth to model the diffusion between homophilous actors within a dynamic social network (e.g. Buskens and Yamaguchi 2002; Centola and Macy 2007; Pfeffer and Carley 2013). In many of these models, agents hold a set of "knowledge bits" that spread throughout a network over time. An agent determines whom to interact with (and hence spread information or beliefs to) based on the similarity of his knowledge to the knowledge of his alters, a form of homophily. The process by which knowledge and belief structures co-evolve with the interactions between agents has been defined as *Constructuralism* (Carley 1991).

One weakness of earlier homophily based diffusion models addressed by Constructuralism is that agents in earlier models tended to be locally omniscient- each agent had a perfect perception of the knowledge bits of his alters. In reality, humans work not with a precise notion of the knowledge of others, but rather with a *perception* of it. Constructuralism addresses this using a mechanism similar to the transactive memory system described by Wegner (1995). In a transactive memory-based simulation, agents update their perception of the knowledge and beliefs of their alters when interactions occur. This perception is then used to determine whom to interact with based on perceived homophily.

Construct is a turn-based, agent-based dynamic-network framework (Carley 1990; Carley, Martin, and Hirshman 2009) that implements the theory of Constructuralism. Formally, each agent *a* in a Construct simulation holds a vector of knowledge bits, $\boldsymbol{k}$, and also a perception of the knowledge of each other agent in the simulation that they are connected to in a modeler-specified social network. On each turn, each agent computes the similarity of his knowledge with all others he is connected to by determining how many knowledge bits he perceives that he shares with them. Formally, the similarity *a* has to another agent *b* can be computed as $\frac{\boldsymbol{k} \,\&\, \boldsymbol{p_b}}{|\boldsymbol{k}|}$ where $\boldsymbol{p_b}$ is *a*'s perception of *b*'s knowledge and both $\boldsymbol{k}$ and $\boldsymbol{p_b}$ are bit vectors. After computing similarity with all others he is connected to, *a* probabilistically selects interaction partners based on relative similarity. He then interacts with a modeler-determined number of alters and exchanges knowledge bits with these actors. This cycle is then repeated for each turn in the simulation. For more details on these mechanisms, we refer the reader to (Carley, 1991).

In addition to knowledge, agents can also hold beliefs. Each *belief* in Construct is represented by a subset of $\boldsymbol{k}$. For example, as we will discuss further below, the violence belief might be linked to certain knowledge facts, while the revolution belief might be associated with others. The *belief network* is then computed using a simple summation of the positively and negatively valenced knowledge bits that a given agent holds for each belief. For further details on how Construct computes this network beyond what is described in this article, we refer the reader to (Lanham, Morgan, and Carley 2014).

Construct has been widely used to examine how ideas diffuse and beliefs change as a function of the underlying social structure in the community. It has been validated several times, most recently by Schreiber and Carley (2012). Validated versions of Construct implemented agent cognition as perceived homophily using a transactive-memory based system. However, while transactive memory moves toward a more realistic simulation of the principle of homophily, a purely transactive-memory based model of agent cognition belittles the fact that humans constantly construct their image of both themselves and those around them at higher-order aggregations than the individual. Mead (1925) argued that humans utilize the concept of the *generalized other,* a perception of the knowledge and beliefs of everyone around us based on what we have learned in previous interactions of those around us and ourselves.

Mead's conceptualization of a single generalized other suggests that humans constantly stereotype the knowledge and beliefs of others based on what we can infer or recall about them. With weaker ties, we rely on observable characteristics and things we can infer or recall about another person (e.g. their occupation) to construct a stereotypical view of their knowledge and beliefs[3]. In truth, it is thus our *constructed perceived homophily* that influences our likelihood of interaction with others in a homophily-based diffusion model, rather than either of the previous mechanisms used to model interaction.

The version of Construct used in the present work has been advanced to incorporate a more cognitively plausible and computationally feasible model of the construction of perceived homophily based on the concept of *constructed perceived homophily*. This model, including full details on mechanisms and the model's faithfulness to socio-cognitive mechanisms, is described in more detail by Joseph et al. (2014). Here, we briefly review the functionality of this tool. To avoid confusion, we will refer to the version of Construct utilized here as *Multi-level Construct (MLC)* in the remainder of the article.

MLC is based on the notion that agents use two *levels* of familiarity to construct the knowledge of a possible interaction partner. The level of cognition an agent uses to construct the knowledge of a possible interaction partner (alter) is based on the strength of the tie between them. Where the tie is stronger, an agent will have a more precise cognitive representation of his alter's knowledge. More specifically, agents who frequently interact will have an *individual-level* perception of the knowledge and beliefs of each other, leading to a model of strong ties that is faithful to the original conceptualization of Constructuralism and a transactive-memory based scheme. With a weaker tie, however, an agent must construct his alter's knowledge via a process of stereotyping.

The agent constructs what he perceives the alter to know from what he knows of the

[3] See (Greenwald and Banaji 1995; Hilton and von Hippel 1996) for reviews of the plethora of social psychology literature addressing stereotyping.

alter's existence in social groups (Tajfel and Turner 1979). Thus, for example, a revolutionary from Egypt, Agent A, who has rarely interacted with someone from Syria, Agent B, may construct what he expects Agent B to know based on the fact that Agent B is from Syria. In the present work, we place agents into groups based on their beliefs about revolution and violence in addition to their nationality. Thus, Agent A may know that Agent B is from Syria, and that he is associated with a social group that is strongly in favor of revolution.

Agents update their perception of social groups as they interact with members of them. Thus, upon interaction, Agent B may pass information to Agent A about how to stage a successful protest. Agent A would then ascribe the knowledge of successful protests to Agent B and to the social groups that Agent B belongs to. Agent A would then be more likely interact in the future with other agents belonging to the same social groups as Agent B, given that he expects that anyone in the group will share his knowledge of how to stage such a protest.

Beyond these mechanisms employed to increase the faithfulness of the model to basic cognitive functions, the mechanisms employed in MLC allow it to run significantly larger simulation models than previous versions of Construct. More specifically, a naïve implementation of agent cognition using only transactive memory would require a matrix of size *O(Number of Agents * Number of Agents * Number of Knowledge bits)* to function. A matrix of this size dominates the memory cost of a social simulation, and updating it can have huge effects on time complexity as well. Previous work shows that MLC has an average space complexity of closer to *O(Number of Agents * Number of Groups * Number of Knowledge Bits)*, which in practice reduces space constraints by an order of magnitude.

4. Simulation Model

MLC, as a generalizable simulation framework, allows for a diverse set of inputs and functionality- for more details, see (Carley et al. 2012). The data and functionality required for a specific research question are detailed within a simulation model, specified to the framework as an XML document. This model can then be run using MLC.

With respect to functionality, our state-revolution model is specified to run for 30 turns. On each turn, any agent can interact with any two others. During each interaction, agents can pass a maximum of two knowledge bits. While agents are allowed to interact with anyone else, they are seeded to have had previous interactions with specific others based on information in the raw data (explained below), making them more likely to interact with these individuals at the beginning of the model. However, as agents learn, they will become more likely to interact with those who share similar knowledge and beliefs, leading to a homophily-based diffusion model as the simulation progresses.

TABLE 1 ABOUT HERE

With respect to the data needed for instantiation, Table 1 gives a description of the meta-network necessary for our state revolution model. It shows that four types of nodes are necessary- beliefs, agents, agent groups and knowledge. We pull agents and knowledge from the raw text data and as modelers specify the form of the beliefs and agent groups. Our model is driven by these latter two nodeclasses, which are aggregate functions of knowledge and agents, respectively. A belief in Construct clusters together a set of knowledge nodes that are relevant to a certain higher order concept. In our state revolution model, we focused on change in the pro-revolution and pro-violence sentiment of the Arab world, and we thus use *violence* and *revolution* as our two beliefs. Beliefs have both positive and negative sentiment- as such, we will have knowledge aligned to both pro and anti sentiments for each of the two beliefs. An agent group clusters together agents that we as modelers believe are associated in such a way that others might form a stereotype about them as a collective. In our model, this will be agents who are from the same country and agents that hold similar beliefs.

Table 1 also shows that we require five networks to instantiate the model. The Agent by Agent network details agents that will have a higher likelihood of interaction at the beginning of the simulation. The Agent by Knowledge network specifies the concepts pulled from the text that each agent is associated with, and consequently those they may share with other agents during the simulation. Both agents and knowledge are also connected to beliefs. The Agent by Belief network specifies an agent's current sentiment for both violence and revolution. The Knowledge by Belief network specifies how different concepts pulled from the text are associated either positively or negatively with the two beliefs. Finally, the agent groups that agents are in are specified in the Agent by Agent Group Network.

Note that there are five other networks that could have been considered that are not used in this model. As beliefs are meant to represent conceptually separate mechanisms driving revolution, the belief by belief network that would represent correlations between these beliefs is not of interest in the present work. Similarly, Construct has no current mechanisms to handle correlations between knowledge nodes (the knowledge by knowledge network) nor correlations between agent groups (the agent group by agent group network). Such relationships may be interesting to model in future work. Finally, as agent groups are implicitly connected to beliefs and knowledge via the agents within these groups, we do not make explicit use of a belief by agent group or a knowledge by agent group network here.

In order to instantiate a model for a particular month, we thus require information on the "who", "what" and "where" of the events occurring in the sixteen countries of interest and relationships between entities in this realm. While this data could have been collected by hand or taken from Subject Matter Experts, the interest of the present work is in a rapid and repeatable process for model instantiation. We thus

turned to publically available data and automated methods to pull the information we required for our state revolution model.

5. Generating the meta-network for instantiation

In this section, we detail how our model is instantiated with the meta-network described above. As opposed to simply pulling nodes and networks naively from the text, we must utilize the concepts of *filtering* and *tuning* to ensure the data is of practical size to run and will provide cogent results. We first discuss these two concepts and then continue to our methodology for meta-network creation.

5.1. Filtering and Tuning

The data used to instantiate a simulation model needs to be both filtered and tuned. By filtering, we mean the process of selecting the subset of nodes that would be objects in the simulation. By tuning, we mean the process of adjusting or adding edges in the relevant networks to support analysis.

Why filter? Clearly, filtering increases the amount of time it takes to prepare the data to instantiate the model. Thus, keeping filtering to the minimum would be valuable from a rapid construction perspective. However, the computational cost of the simulation must also be considered – the larger the number of actors and knowledge bits simulated, the longer it takes for the simulation to run. Thus, from a purely time-savings perspective there is a tradeoff in time spent filtering versus time spent running the simulator. A small amount of filtering can have large gains in reducing simulation time. Filtering also aids in reducing data bias and increasing the prevalence of relevant data. Recall that at its core the simulation being used examines the diffusion of pro-revolution and pro-violence beliefs within the country in question. To that end, the presence of actors and knowledge not relevant to revolution or violence during the Arab Spring is not only not relevant, it is also distracting from the focus.

The reason such nodes are present in the first place has to do with the nature of the news. Much of the news is associated with events, activities, and actors that have little direct relation to the country in question. An example would be articles about foreign soccer stars who had just played or were about to play a team in the country in question. Another example would be a topic such as US gas prices, which might occur in an article about how violence in the country in question could impact US gas prices. Filtering helps to remove these extraneous nodes; if it can be done in a rapid, principled and repeatable fashion it therefore enables both more rapid analysis and more accurate and focused analysis.

Why tune? Like filtering, tuning increases the time to prepare the data to instantiate the model. Furthermore, tuning does not speed up model processing. However it is necessary to enable the model to be used at all. We use the concept of tuning to infer agent's initial beliefs and their membership in social groups. The news rarely

directly provides input on what stance an actor takes toward the beliefs in a particular model and rarely distinguishes clear group memberships. But to use the model, both of these connections are needed. Tuning thus makes it possible to make inferences about the higher-order social structures, like beliefs and groups. If it can be done in a rapid, principled and repeatable fashion tuning supports both data augmentation with secondary sources and inference of missing data, thus enabling more detailed and nuanced simulations to be run.

5.2.Meta-network creation

Figure 3 ABOUT HERE

Figure 3 shows the seven-step process we use to generate a meta-network from a set of articles. In the figure, there are five different types of boxes. All black boxes represent raw data input to the model, either by the modeler (the seed topics and list of Westerners, as discussed below) or from the LexisNexis articles. White boxes with a black outline represent processes that manipulate data. Boxes with grey outlines represent unfiltered or partial representations of nodeclasses, while the grey-filled colored boxes represent the final nodeclasses in the meta-network. Finally, boxes with dashed lines around them represent the networks in the meta-network. In addition to the boxes in Figure 3, arrows labeled with step numbers are drawn to indicate the movement of data through the generation mechanism. Below, we detail each of the seven steps represented in the diagram.

5.2.1. Step 1: Obtaining the raw agents, knowledge and countries from the text

We are provided with two sources of information from the set of newspaper articles pulled from LexisNexis- the raw text of the articles and the terms used to index each article. From the indexed terms, we obtain the topics and countries being discussed in any given article. Note that topics are distinct from knowledge, as we will derive the actual knowledge nodeclass used in the simulation from the topics we discover. Because LexisNexis did not include key actors from the Arab Spring (for example, prime ministers of several of the countries studied) in the indexing system, it was necessary to extract the agents discussed in each article directly from the text. In order to do so, we rely on the Stanford Named Entity Recognizer (NER), which uses a Conditional Random Fields approach to pull named entities from a given text (Finkel, Grenager, and Manning 2005). Though the model is trained only on news from the United States and England, we find qualitatively that it provides a reasonable collection of persons involved in the Arab Spring from English news sources.

5.2.2. Step 2: Filtering of agents

There are four substeps in the filtering of the agent nodeclass: noise removal, de-duplication, the removal of agents not associated with the countries of interest and

the removal of agents not associated with the topics of interest. To reduce noise in the data, we first remove any names discovered by the NER that were of length one (e.g. *Bill*) and any names longer than length five. These names rarely referred to actual people of interest.

De-duplication is completed according to two heuristics. First, we combine into a single actor any names returned by the NER that have a string edit distance of less than four. We find that a distance of three generally suggests alternate spellings of Arabic names by Western journalists, while increasing the threshold any further results in a sharp decrease in the number of agents identified (i.e. too many combinations). Second, we merge into a single agent all names pulled by the NER where one name is *approximately consumed by* another. In order to do so, we take the smaller of the two names (or, where length is equal, either name) and determine the proportion of space-delimited terms in the longer name that are also in the shorter name. If this proportion is greater than or equal to some threshold, then we assume the two names refer to the same actor. In the present work, we chose a threshold of .75, meaning that, for example, *Hillary Clinton* and *Hillary Rodham Clinton* would not be merged, while *President Barack Hussein Obama* and *President Barack Obama* would be merged. The selection of this threshold was used because qualitative exploration of the data suggested that the most frequent issues surrounding names being approximately consumed by others occurred when Western journalists only included "first, middle and last names" of Arabic names and, similarly cases where actors' titles are included with their first, middle and last names in some articles and not in others. In both cases, a threshold of .75 is effective in reducing the number of agents in our dataset.

Our heuristics fall under the broader umbrella of query correction, for which there exist a variety of both formal (e.g. Li, Duan, and Zhai 2012) and informal[4] techniques. More formal, intricate methodologies show promise in being able to further deduplicate terms produced by the NER. For example, we do not consider creating semantic cross-references; we make no attempt to identify Hosni Mubarak and "the president of Egypt" as referring to the same actor. However, our relatively greedy heuristics provide a starting point from which to simulate while also being efficient to implement and run on large corpora.

After noise removal and deduplication, we have a partial set of agents. We wish to further filter this set, however, by retaining only agents associated with one of the sixteen countries of interest and at least one of the topics we use in the model. To associate each agent with countries, we use co-occurrence information, determined by calculating the number of times a given agent was mentioned in an article indexed by a given country. We associate each agent with a single country, the one that they co-occurred most frequently with. Thus, for example, Hosni Mubarak, if mentioned in ten articles indexed by Egypt and three articles indexed by the United

[4] E.g. the FuzzyWuzzy python module, https://github.com/seatgeek/fuzzywuzzy, used by companies like StubHub

States, would be associated with Egypt. Agents are also associated to knowledge nodes through co-occurrence, but in this case with the topics these knowledge nodes represent. This process, and the final step in filtering the Agent nodeclass, is detailed in Step 4 (Section 5.2.4).

5.2.3. Step 3: Generating the Knowledge by Belief Network and the Knowledge nodeclass

TABLE 2 ABOUT HERE

When generating the knowledge by belief network, we must have some indication of the relationship of each topic discussed to a particular valence (positive or negative) of the two beliefs that we use in the model. We then will use the strength of each topic's association with a given belief valence to generate a set of knowledge nodes for each topic. In order to determine the valence of topics, we first define a subset of the index terms in the LexisNexis database that can generally be associated with a valence on the two beliefs of interest. These *seed topics* are not specific to the Arab Spring per se, but are rather concepts for what generally may bring about or result from pro (anti) revolution (violence) sentiments within a population. Table 2 defines the ten *seed topics*, selected via iterative coding, associated with positive and negative sentiments used in the present work along the revolution and violence beliefs.

Using these seeds, we also wish to uncover a set of additional topics that are also associated with our beliefs. In other words, we would like to *expand* the seed topics with terms that are related in some way in our set of articles. The procedure we use to do so is relatively straightforward and is common in both sentiment mining (see Pang and Lee 2008, p. 27-28) and automated query expansion (AQE), the process of uncovering terms related to a user's query in order to provide them with more pertinent results (Carpineto and Romano, 2012). More complex approaches to AQE tend to incorporate contextual information about documents in which terms are found (e.g. the term's position in the document). Because we work only with index terms that may not even be in the document, we choose to utilize a more straightforward methodology that can be carried out based solely on co-occurrence information.

Our method assumes that topics co-occurring frequently with one or more of the terms provided in Table 2 and infrequently without any of these terms will be most related to our two beliefs. We refer to the lists of topics defined in the columns of Table 2 as $\boldsymbol{r}_+$, $\boldsymbol{r}_-$, $\boldsymbol{v}_+$ and $\boldsymbol{v}_-$, respectively from left to right. For each topic in $\boldsymbol{r}_+ \cup \boldsymbol{r}_- \cup \boldsymbol{v}_+ \cup \boldsymbol{v}_-$, we can construct a *topic-article vector (TAV),* $\boldsymbol{t}^{\boldsymbol{topic\_name}}$, that has $|\boldsymbol{A}|$ entries, where $\mathbf{A}$ is the set of all articles in the given month. The *i*th entry of a given TAV, $t_i^{\boldsymbol{topic\_name}}$, is a binary value representing whether or not the topic appeared in the *i*th article of the given month.

For each topic not in $\boldsymbol{r}_+ \cup \boldsymbol{r}_- \cup \boldsymbol{v}_+ \cup \boldsymbol{v}_-$ we can also compute a TAV. We then calculate the similarity of each topic's vector to all topics in Table 2 using a weighted version of the F1 metric similar to the one used by Raina, Ng, and Koller (2006). The F1 metric (equivalent to the Dice coefficient[5]) measures the extent to which a term appears in an article if and only if a second term appears. Thus, the F1 metric as used here can be thought of as the extent to which one topic, represented by the TAV x, occurs only in articles where the other topic, represented by the TAV y, also occurs. Equation 1 specifies this mathematically, giving the formula for our weighted F1 (WF1) similarity metric.

$$WF1(x,y) = \log(|xy|) * \frac{2\frac{|xy|}{|x| * |y|}}{\frac{1}{|x|} + \frac{1}{|y|}} \quad (1)$$

FIGURE 4 ABOUT HERE

Figure 4 provides a visualization of the WF1 between two topics important to the Egyptian revolution and each of the terms in Table 2. The set of articles used is all those written in January 2011, the month violence broke out in Egypt. On the y-axis, we observe the outcome of the similarity scores for *food prices* and *internet social networking* to each seed topic. As is clear, food prices are highly associated with positive valences along our revolution belief. A recent article from the Economist[6] suggests that this association is far from trivial, and that "food has played a bigger role in the upheavals than most people realise". Similarly, as previous work has suggested (Papacharissi and de Fatima Oliveira 2012), social networking online played an important role in the onset of revolution in Egypt. These two examples give anecdotal evidence that our approach provides a useful mechanism to quickly pull relevant topics from newspaper articles.

Using our similarity metric, the valence of each topic can be defined via some combination of its WF1 score with the positive and negative valence terms we specified. We choose here to sum the scores for each term, thus treating each seed topic as an independent indicator of the relevance of a topic to a particular valence on our two beliefs. As an example, Equation 2 gives the formula used to calculate r, the valence along the revolution belief for the TAV $\boldsymbol{t}^{\boldsymbol{food}}$.

$$r = \sum_{topic \in \boldsymbol{r}_+} WF1\left(\boldsymbol{t}^{\boldsymbol{topic}}, \boldsymbol{t}^{\boldsymbol{food}}\right) - \sum_{topic \in \boldsymbol{r}_-} WF1\left(\boldsymbol{t}^{\boldsymbol{topic}}, \boldsymbol{t}^{\boldsymbol{food}}\right) \quad (2)$$

[5] Trivially, see http://brenocon.com/blog/2012/04/f-scores-dice-and-jaccard-set-similarity/ for a derivation

[6] http://www.economist.com/node/21550328

FIGURE 5 ABOUT HERE

Most topics align close to zero along both beliefs. In order to perform filtering, we removed all topics inside of two standard deviations from the mean for each belief. Importantly, topics that have a high valence on one belief and not another belief are set to zero on the opposing belief. Figure 5 gives a display of the distribution of valences of all topics used in the model for January 2011, showing a gap near zero on both axes where uninteresting topics were removed or moved to zero along one of the beliefs.

Having associated topics to beliefs, we now must determine how to move from the abstraction of a topic to the concept of a knowledge node that is central to our simulation model. While a one-to-one mapping from topic to knowledge node would be a straightforward solution, we opt for a slightly more complex model to account for the fact that some topics have significantly stronger valence than others. To move from topics to knowledge bits, we allow extra knowledge bits for a given topic based on a logarithmic scaling of the similarity weights. Thus, while in most cases a topic will be represented by a single knowledge bit, some topics that have a disproportionately high valence, such as *internet social networking*, would be represented with multiple knowledge nodes, each of which has a valence of +1 in favor of revolution. This relationship creates the knowledge by belief network.

5.2.4. Step 4: Generating the Agent by Knowledge network and finalizing the Agent nodeclass

The agent by knowledge network actually serves two functions. The first determines the number of knowledge bits for a particular topic that an agent knows. Where a topic has only a single knowledge bit, an agent will be connected to it in the agent by knowledge network if the two co-occur in any article in the given month. If the topic has more than one associated knowledge node (bit), the agent will only be connected to a larger number of bits if he co-occurs with the topic that number of times. Thus, an agent co-occurring with the topic *internet social networking* three times would be given three of relevant knowledge bits, while an agent co-occurring with the term five times would obtain five of the knowledge bits associated with this topic (if five such bits existed). Given the heavy-tailed distribution of co-occurrences between agents and relevant topics in general, this heuristic provides a model suitable for instantiation.

The second function of the agent by knowledge network utilized concerns the likelihood that an agent would transmit any given knowledge bit to an alter on a given interaction. In order to obtain a probability distribution across knowledge bits, we first create a distribution across topics for each agent based on co-occurrences. We rescale agent's associations with all topics to sum to one, giving a probability of transmitting any given topic. An agent will then transmit any given

knowledge bit within a particular topic with equal likelihood. Thus, if an agent had a 50% chance of transmitting information about *internet social networking* on any given turn and knew two of the relevant knowledge bits, he would have 25% chance of transmitting either of these during interaction.

5.2.5. Step 5: Generating the Agent by Belief network

To create groups of agents according to their beliefs, we first must create a representation of agents' beliefs. This is done by summing the valences of all topics an actor co-occurred with for each belief. Mathematically, we can represent the creation of the Agent by Belief network as the matrix multiplication ***I(AT)*TB***. The term I(**AT**) represents the binarized form of the agent by topic network, where the index $\mathbf{AT_{a,t}}$ is 1 if Agent *a* co-occurred in any article with Topic *t*, and is zero otherwise. The term **TB** simply represents the topic by belief network, where the value $\mathbf{TB_{t,b}}$ is found via Equation 2.

5.2.6. Step 6: Generating Agent groups and the Agent by Group Network

We consider three types of agent groups in the model. First, we define a *Westerner* group, which is used to define agents that were associated with one of the sixteen Arab Spring countries in Step 1 but who are known to be actors from Western nations. In order to do so, we defines partial names, e.g. "Bush" and "Obama", which represent these well known actors and then place any agent whose name contains the terms provided into the "Westerner" group. These agents were not included in either of the other two grouping methodologies described below.

The second type of groups are based on country. Having aligned each agent with a country in Step 1, these groups are trivial to create. Finally, we create groups based on belief homophily at the global and per-country levels. From the Agent by Belief network obtained in Step 5, we have a two-dimensional representation of agents, which can be considered a latent social space (McPherson and Ranger-Moore 1991). Krivitsky et al. (2009) suggest that an appropriate technique for clustering actors in a latent social space is some form of model-based clustering. Here, we use a Gaussian mixture model to find clusters of agents in the latent belief space.

All clustering was done using the *mclust* package (Fraley et al. 2012) in R (R Core Team, 2012) with a covariance matrix allowing for variable volume, shape and orientation of the clusters. We determine the optimal number of clusters by taking the model with the best BIC, which gives a stricter penalty for "adding" another cluster than the AIC or other similar best-fit statistics (Wasserman, 2003). Thus, the number of agent groups is variable, based on a clustering of the agents that best fits the data. However, we only consider a maximum of twenty clusters due to the computational costs associated with attempting groupings at higher levels.

FIGURE 6 ABOUT HERE

Figure 6 shows the best clustering for January 2011 across all agents associated with the country Egypt. Agents are aligned by their revolution belief (the x-axis) and their violence belief (the y-axis). The visualization was created using the network analysis tool ORA (Carley et al. 2012). In the figure, the color of the underlying nodes represents their association with a particular cluster. The origin can be determined by observing the large clustering of agents near the bottom right of the figure, showing that most agents are relatively neutral along both beliefs. Figure 6 also shows that variances of the clusters near the origin are thus much smaller, indicating that groups of agents who have more extreme beliefs must capture a larger subset of the latent space to include a similar number of agents. Finally, we note that most extreme beliefs were aligned heavily with anti-revolution. These entities were nearly all government officials.

5.2.7. Step 7: Generating the Agent by Agent Network

The final step is to create the Agent by Agent network. This step must be completed last, as our final agent nodeset is not determined until we generate both the agent by topic network and the agent by country network (a subset of the Agent by Group network). Once we obtain the final set of agents, the Agent by Agent network is uncovered simply using co-occurrence data. This network is used to seed agents with possible initial interaction partners. More specifically, in *MLC* (as in real life- see Joseph et al., 2014) agents are more likely to interact with individuals they have recently interacted with. As noted above, this is modeled by giving agents an individual-level perception of the knowledge of this specific interaction partner, as opposed to perceiving this alter as part of a group. Over time, this individual-level perception is forgotten, and the alter again becomes just another member of a group.

Thus, by using the Agent by Agent network to seed agents with previous interactions, agents begin the simulation with these more concrete cognitive representations of any other alter they co-occurred with in an article, analogous to a stronger tie between these actors. While this does not guarantee that these two agents will interact during the simulation, it does make it more likely. This matches our intuition that actors mentioned in the same newspaper article are more likely to have interacted in real life than individuals not mentioned together, but still somewhat unlikely to have done so.

6. Outcome Description

Overall, we simulate 20 months of data. Eight replications are performed for each month of data, each with slightly different parameterizations of MLC. With eight replications of each month this is a total of 160 simulation runs, each of which simulates between 7000 and 13000 agents. Parameter differences were motivated by previous work (Joseph et al., 2014). As output, we collected the Agent by Belief network from the last time period for each run and subtract the initial agent by belief network created in Step 5. We then sum the resulting matrix across all agents associated with each of the sixteen countries we study here, giving us the *change in*

belief of the agents associated with each country from the beginning to the end of the simulation. Because we found no interesting differences across parameterizations, we use the mean of these replications for point estimates. Note, however, that we use all replications individually to compute inter-quartile ranges (IQR), as described below.

Because our focus is on initial change points that might indicate a revolution will occur in a particular country, we translate change in belief into a binary predictor which indicates whether or not revolution is possible in a particular country for a particular month. In order to do so, we leverage our expectation that revolution and violence beliefs in a country on the verge of revolution would be 1) noticeably different than the values of previous months for that same country and 2) noticeably different than the values of the same month for all other countries.

Because beliefs across countries were not normally distributed, we adopt a non-parametric approach. To determine whether or not a single country's revolution and/or violence beliefs were noticeably different than previous months, we consider all twenty-four simulation runs for the present month and the two previous months. For each country and each belief, we compute the inter-quartile range of the simulation outputs using R, which determines the IQR using Definition 7 in (Hyndman and Fan, 1996). We refer to these ranges as *intra-country ranges*[7]. If a country's belief value for a particular month is outside of its intra-country range, the value is said to have noticeably changed. Similarly, to determine whether or not a country's revolution and/or violence belief is noticeably different from all other countries in that month, we compute the IQR of each belief for all countries in that month. We refer to this statistic as the *inter-country range*.

If the changes in a country's violence and revolution beliefs are both outside of their respective *intra-country range* and *inter-country range* in a given month, we determine that revolution is likely. For the purposes of predicting revolution, we use the first such month as our binary predictor of revolution. Thus, we obtain as a final outcome, for each country, zero or one month in which the model predicts revolution has become likely.

7. Results

FIGURE 7 ABOUT HERE

Figure 7 shows the mean change in the revolution and violence beliefs for each country and each month. The grey line represents change in the revolution belief

[7] So, for example, the intra-country range for Egypt's change in violence belief in January of 2011 would be computed as the IQR of Egypt's change in violence belief from all runs covering January of 2011 or November or December of 2010.

and the black line represents violence. Vertical black lines at each month represent the *inter-country ranges* and grey lines at each month represent the *intra-country ranges*. Large black dots indicate the month in which the model predicts that a revolution is likely. Note that the magnitude of the scale for each country is unique in order to show variation on a month-by-month level for each nation.

Before discussing the relation of model predictions to real-world findings, obvious features of the results are discussed. First, Figure 7 shows that the magnitude of change for the two beliefs closely mirrored each other. This suggests that agents who were frequently mentioned in the context of one belief were often mentioned in the context of the other as well. However, while violence tended to increase when moving outside of a baseline range, revolution sentiment tended to *decrease*. This contradicts our a priori belief that revolution sentiment would increase when protests and revolution occurred in the Arab world. Instead, results and qualitative analysis of the data suggest that when English-speaking journalists discussed the Arab Spring, the focus was on what was being done to construct solutions to the violence that was occurring. These steps often involved actions related to the anti-revolution seed topics listed in Table 2. Because of this, we adopted our mechanism for predicting revolution to make predictions based on the absolute value of beliefs (and the respective inter-quartile ranges). Thus, revolution was indicated when the revolution belief was noticeably negative and the violence belief was noticeably positive.

TABLE 3 ABOUT HERE

Table 3 shows, for each country, the month and year that the reigning government was overthrown by revolution (or None if this never occurred) and the month our model predicted revolution first became likely. The model correctly raises no sign of revolution in six nations only tangentially associated with the Arab Spring. The model also correctly predicts that revolution would occur in the four nations where governments were overthrown during the period of study. Importantly, however, the model does so with varying levels of temporal accuracy. Model predictions of revolution preceded actual overthrow by six months, four months and zero months in Libya, Egypt and Tunisia respectively. In Yemen, the model first predicted a revolution would occur two months *after* the government was overthrown, a prediction we consider to be incorrect.

The model also predicts revolution in six nations where governments were not overthrown. In two of these nations, revolution indeed occurred but, as of the writing of this article, has not been successful in overthrowing the government in place. In Bahrain, though the government was not overthrown, significant attempts at revolution were made before being crushed by the reigning regime[8]. In Syria, though the government has not yet been overthrown, a civil war continues that

[8] See, e.g., http://www.guardian.co.uk/world/interactive/2011/mar/22/middle-east-protest-interactive-timeline.

began in late March of 2011. Thus, though model output is here compared to successful revolutions, these two nations represent model output that correctly predicted the rise of revolution.

Finally, the model predicts that revolution was likely in four nations that have seen little coverage in connection with the Arab Spring. All four of these countries represent nations with strong diplomatic functions in the region, and thus were the foci of journalists covering the broader impacts of the Arab Spring on the region. Furthermore, the ongoing conflicts between Iraq, Iran and the United States during the timeperiod of study provided additional coverage of these nations not related to the Arab Spring. Consequently, we observe that the model could be improved in the future by learning to differentiate the context in which the topics in Table 2 are covered.

8. Conclusion

We have presented an approach that enables a simulation model to be instantiated in a semi-automated fashion. The core advantage is this enables model reuse, and supports improved validation. We illustrated this approach using the Arab Spring. We created a state level revolution model using the Construct framework that we instantiated and ran across twenty different months. The suite of simulations from start to finish the model took less than a week to run (or approx. 6 hours per simulated month start to finish), albeit on powerful hardware[9]. At the technological level our results indicate the value of this semi-automated approach to instantiating an agent-based dynamic network.

On the strict prediction task of determining when a successful revolution would occur in a country before it actually occurred, our model achieved 75% recall at the expense of 30% precision. On the looser scale of correctly predicting significant revolutionary activities in a country, the model does much better, achieving 100% recall and 60% precision. At the theory level, our state-revolution model thus appears to be useful for predicting revolutionary activity.

Model accuracy could likely be improved via model tuning, in particular, the modification of parameters to MLC and changes to the seed topics to provide more complete coverage. While such tuning would need to be carefully conducted in order to avoid overfitting, the issue of topic coverage is critical. In other recent work, we found that immediately prior to the initial revolutionary event the complexity of the topics (number of topics and their interconnectivity in a topic network) and the number of actors of interest actually increases (Pfeffer and Carley 2012). Thus, we suggest that accuracy could be further improved by a more extensive extraction of data from the news articles, and accounting for the inherent

[9] All replications for a single month were run in parallel using 8 cores of a 60 core machine with a 250 GB SSD drive and 120 GB of RAM.

non-linearity due to topic interconnectivity. Furthermore, as we focus only on English-language newspaper data, results could also likely be improved by incorporating data from both foreign language newspapers and mediums on which information is disseminated more rapidly, such as Twitter.

Though some model tuning did occur over the course of model development, the focus of the present work was the process by which the model was instantiated and utilized to make predictions. To this end, future efforts to improve the techniques described here are also important. For example, efforts to incorporate less heuristically based deduplication approaches are needed, as the wholly unsupervised approach we take to entity recognition is likely to provide a moderate level of false de-duplications (i.e. combining two people into one agent) and duplicates. Additionally, determining a methodology to update results from each month in a Bayesian fashion using the priors from the previous months seems like an appropriate methodology to construct more accurate predictions. Finally, while our approach was intended to be domain agnostic, future work should consider how well the methods described really do transfer to new domains.

Irrespective of the need for these future efforts, the present work bodes well for the field of simulation as a real policy tool as it provides the methodological groundwork for methods of making models re-usable through reduced effort in instantiation across different contexts.

9. Acknowledgements

This work was supported in part by the Air Force Office of Sponsored Research, FA9550-11-1-0179 and the Office of Naval Research through a Minerva N000141310835. Additional support was provided by the center for Computational Analysis of Social and Organizational Systems and the Institute for Software Research at Carnegie Mellon University. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the National Science Foundation, or the U.S. government.

References

Baldwin, Mark W. 1992. “Relational Schemas and the Processing of Social Information.” *Psychological Bulletin* 112 (3): 461–484. doi:10.1037/0033-2909.112.3.461.

Buskens, Vincent, and Kazuo Yamaguchi. 2002. “A New Model for Information Diffusion in Heterogeneous Social Networks.” *Sociological Methodology* 29 (1) (December 17): 281–325.

Campante, Filipe R., and Davin Chor. 2012. “Why Was the Arab World Poised for Revolution? Schooling, Economic Opportunities, and the Arab Spring.” *The Journal of Economic Perspectives* 26 (2): 167–187.

Carley, K. M. 2002. “Smart Agents and Organizations of the Future.” *The Handbook of New Media* 12: 206–220.

Carley, Kathleen. 1990. “Coordinating for Success: Trading Information Redundancy for Task Simplicity.” *Technical Report, Institute for Software Research, Carnegie Mellon University*.

———. 1991. “A Theory of Group Stability.” *American Sociological Review* 56 (3) (June 1): 331–354.

Carley, Kathleen M, David T Filonuk, Kenny Joseph, Michael Kowalchuck, Michael J Lanham, and Geoffrey P Morgan. 2012. “Construct User Guide.” *Technical Report, Institute for Software Research, Carnegie Mellon University*.

Carley, Kathleen M, Michael K Martin, and Brian R Hirshman. 2009. “The Etiology of Social Change.” *Topics in Cognitive Science* 1 (4) (June 26): 621–650.

Carley, Kathleen M, Jürgen Pfeffer, Jeff Reminga, Jon Storrick, and Dave Columbus. 2012. “ORA User’s Guide 2012”. DTIC Document.

Carley, Kathleen M., Geoffrey Morgan, Michael Lanham, and Jürgen Pfeffer. 2012. “Multi-Modeling and Socio-Cultural Complexity: Reuse and Validation.” *Advances in Design for Cross-Cultural Activities* 2: 128.

Carpineto, Claudio, and Giovanni Romano. 2012. “A Survey of Automatic Query Expansion in Information Retrieval.” *ACM Comput. Surv.* 44 (1) (January): 1:1–1:50.

Centola, Damon, and Michael Macy. 2007. “Complex Contagions and the Weakness of Long Ties.” *American Journal of Sociology* 113 (3) (November 1): 702–734.

Diesner, Jana, and Kathleen M. Carley. 2008. “Conditional Random Fields for Entity Extraction and Ontological Text Coding.” *Computational and Mathematical Organization Theory* 14 (3): 248–262.

Diesner, Jana, Terrill L. Frantz, and Kathleen M. Carley. 2005. “Communication Networks from the Enron Email Corpus ‘It’s Always About the People. Enron Is No Different.’” *Comput. Math. Organ. Theory* 11 (3) (October): 201–228.

Eisenstein, Jacob, Brendan O’Connor, Noah A. Smith, and Eric P. Xing. 2010. “A Latent Variable Model for Geographic Lexical Variation.” In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, 1277–1287. EMNLP ’10. Stroudsburg, PA, USA: Association for Computational Linguistics.

Epstein, Joshua M., and Robert Axtell. 1996. *Growing Artificial Societies: Social Science from the Bottom up*. Brookings Institution Press.

Finkel, Jenny Rose, 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*, 363–370. ACL ’05. Stroudsburg, PA, USA: Association for Computational Linguistics.

Fraley, Chris, Adrian E. Raftery, T. Brendan Murphy, and Luca Scrucca. 2012. “Mclust Version 4 for R: Normal Mixture Modeling for Model-Based Clustering, Classification, and Density Estimation”. Technical Report.

Gilbert, Nigel. 2007. *Agent-Based Models*. Thousand Oaks, CA: Sage Publications Inc.
Greenwald, Anthony G., and Mahzarin R. Banaji. 1995. "Implicit Social Cognition: Attitudes, Self-Esteem, and Stereotypes." *Psychological Review* 102 (1): 4–27.
Hamdy, Naila, and Ehab H. Gomaa. 2012. "Framing the Egyptian Uprising in Arabic Language Newspapers and Social Media." *Journal of Communication* 62 (2):
Hilton, James L., and William von Hippel. 1996. "Stereotypes." *Annual Review of Psychology* 47 (1): 237–271.
Hyndman, Rob J., and Yanan Fan. 1996. "Sample Quantiles in Statistical Packages." *The American Statistician* 50 (4): 361–365.
Joseph, Kenneth, Geoffrey P Morgan, and Kathleen M Carley. 2014. "On the Coevolution of Stereotype, Culture and Social Relationships." *Social Science Computer Review*.
Kossinets, Gueorgi, and Duncan J. Watts. 2009. "Origins of Homophily in an Evolving Social Network1." *American Journal of Sociology* 115 (2) (September): 405–450.
Krivitsky, Pavel N., Mark S. Handcock, Adrian E. Raftery, and Peter D. Hoff. 2009. "Representing Degree Distributions, Clustering, and Homophily in Social Networks with Latent Cluster Random Effects Models." *Social Networks* 31 (3): 204–213.
Lanham, Michael J., Geoffrey P. Morgan, and Kathleen M. Carley. 2014. "Social Network Modeling and Agent-Based Simulation in Support of Crisis De-Escalation." *IEEE Transactions on Human-Machine Systems*.
Lazarsfeld, PF, and RK Merton. 1954. "Friendship as a Social Process: A Substantive and Methodological Analysis." In *Freedom and Control in Modern Society*, 18–66. Van Nostrand.
Li, Yanen, Huizhong Duan, and ChengXiang Zhai. 2012. "A Generalized Hidden Markov Model with Discriminative Training for Query Spelling Correction." In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 611–620.
Licoppe, Christian. 2004. "Connected Presence: The Emergence of a New Repertoire for Managing Social Relationships in a Changing Communication Technoscape." *Environment and Planning D: Society and Space* 22: 135–156.
Martin, Michael K., Juergen Pfeffer, and Kathleen M. Carley. 2013. "Network Text Analysis of Conceptual Overlap in Interviews, Newspaper Articles and Keywords." *Social Network Analysis and Mining*: 1–13.
McPherson, J. Miller, and James R. Ranger-Moore. 1991. "Evolution on a Dancing Landscape: Organizations and Networks in Dynamic Blau Space." *Social Forces* 70 (1) (September 1): 19–42.
McPherson, M., L. Lovin, and J. Cook. 2001. "Birds of a Feather: Homophily in Social Networks." *Annual Review of Sociology* (1): 415–444.
Mead, George Herbert. 1925. "The Genesis of the Self and Social Control." *International Journal of Ethics* 35 (3) (April 1): 251–277.
Pang, Bo, and Lillian Lee. 2008. "Opinion Mining and Sentiment Analysis." *Foundations and Trends in Information Retrieval* 2 (1-2): 1–135.

Papacharissi, Zizi, and Maria de Fatima Oliveira. 2012. “Affective News and Networked Publics: The Rhythms of News Storytelling on #Egypt.” *Journal of Communication* 62 (2): 266–282.

Pfeffer, Jürgen, and Kathleen M Carley. 2013. “The Importance of Local Clusters for the Diffusion of Opinions and Beliefs in Interpersonal Communication Networks.” *International Journal of Innovation and Technology Management* 10 (5).

Pfeffer, Jürgen, and Kathleen M. Carley. 2012. “Rapid Modeling and Analyzing Networks Extracted from Pre-Structured News Articles.” *Computational and Mathematical Organization Theory* 18 (3): 280–299.

R Core Team. 2012. *R: A Language and Environment for Statistical Computing*. Vienna, Austria. http://www.R-project.org/.

Raina, Rajat, Andrew Y. Ng, and Daphne Koller. 2006. “Constructing Informative Priors Using Transfer Learning.” In *Proceedings of the 23rd International Conference on Machine Learning*, 713–720.

Rogers, Everett M. 2003. *Diffusion of Innovations, 5th Edition*. Simon and Schuster.

Schreiber, C., and K. M. Carley. 2012. “Validating Agent Interactions in Construct Against Empirical Communication Networks Using the Calibrated Grounding Technique.” *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans* PP (99): 1 –9.

Tajfel, Henri, and John C. Turner. 1979. “An Integrative Theory of Intergroup Conflict.” In *The Social Psychology of Intergroup Relations*, W Austin & S. Worche, 33–47. Monterey, CA: Brooks/Cole.

Wasserman, Larry. 2003. *All of Statistics: A Concise Course in Statistical Inference*. Springer.

Wegner, D. M. 1995. “A Computer Network Model of Human Transactive Memory.” *Social Cognition* 13 (3): 319–339.

Figures

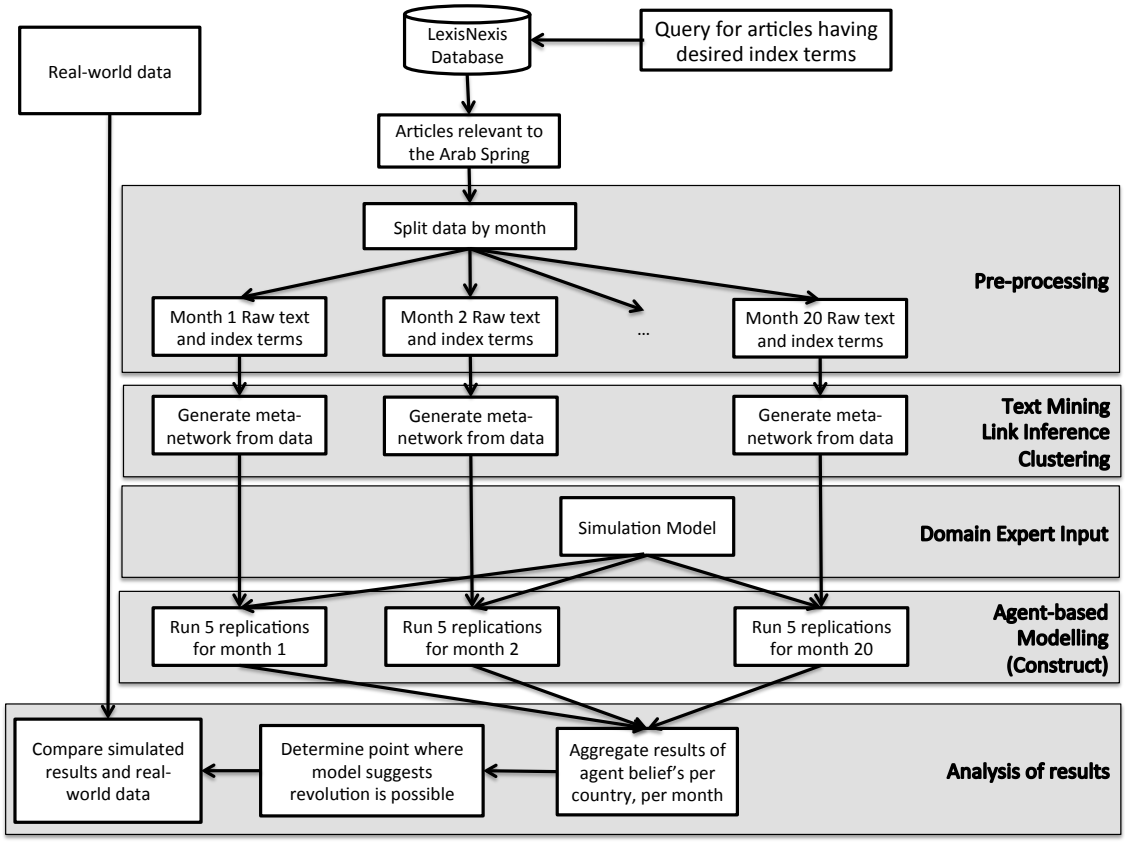

Figure 1- Graphical overview of the methodology presented in the article

Figure 2 – A map of the countries considered in the present article. The legend shows country names, used as search terms used to obtain the sample of newspaper articles from Major News Publications from the *LexisNexis* database for the present study.

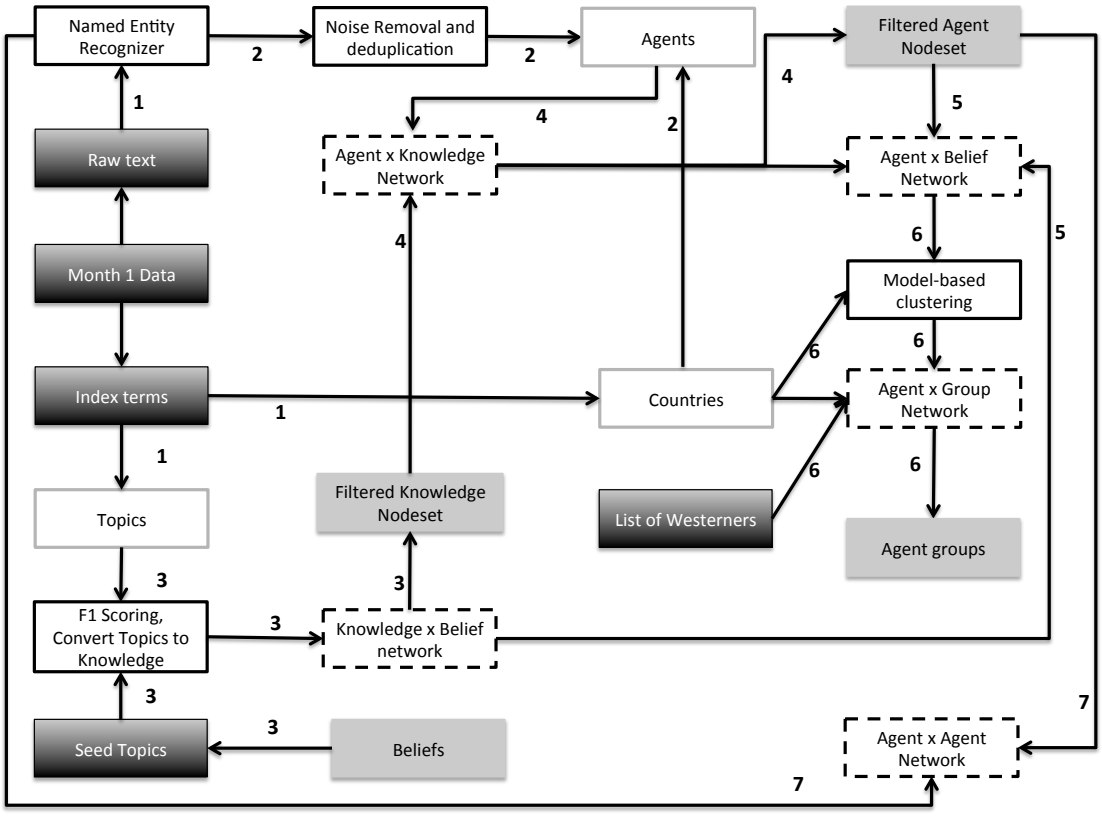

Figure 3- Graphical overview of process used to go from raw newspaper data to the meta-networks used for instantiation of the model

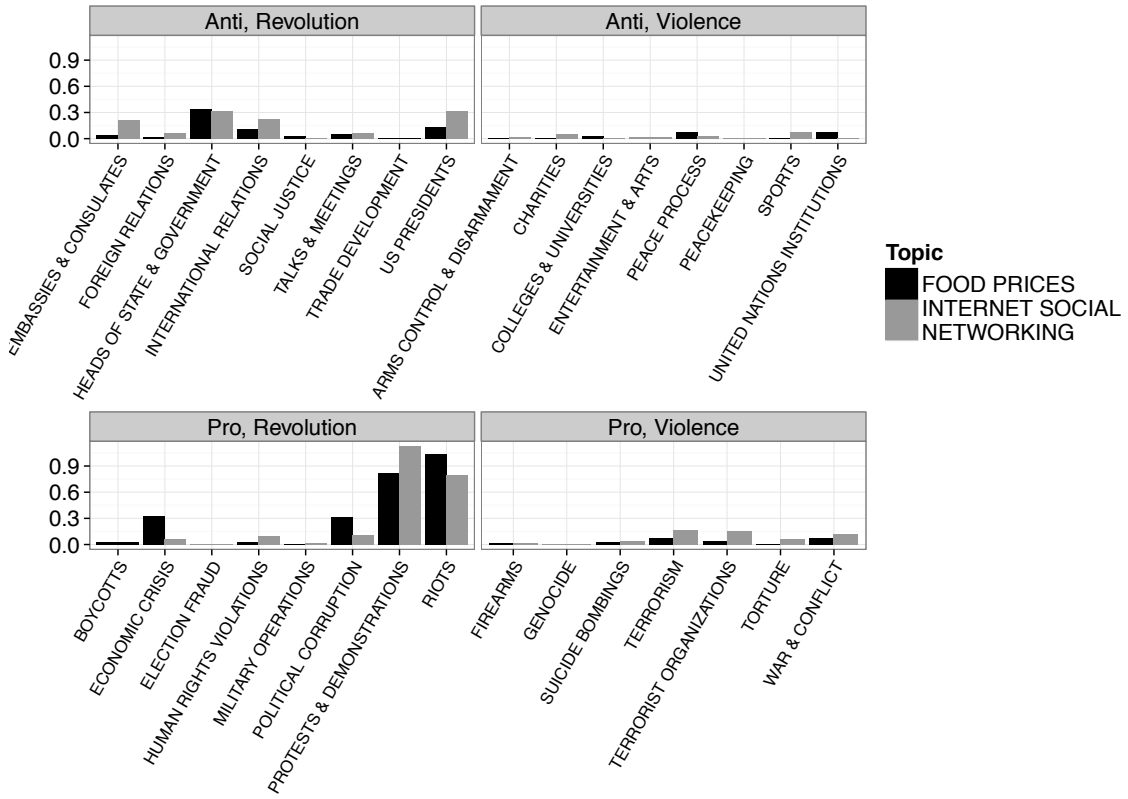

Figure 4 – A bar plot of the WF1 similarity score (y-axis) between the food prices (turquoise bars) and internet social networking (red bars) topics with each seed topic for positive (pro) and negative (anti) sentiments for the violence and revolution beliefs (terms along the x-axis) during January of 2011.

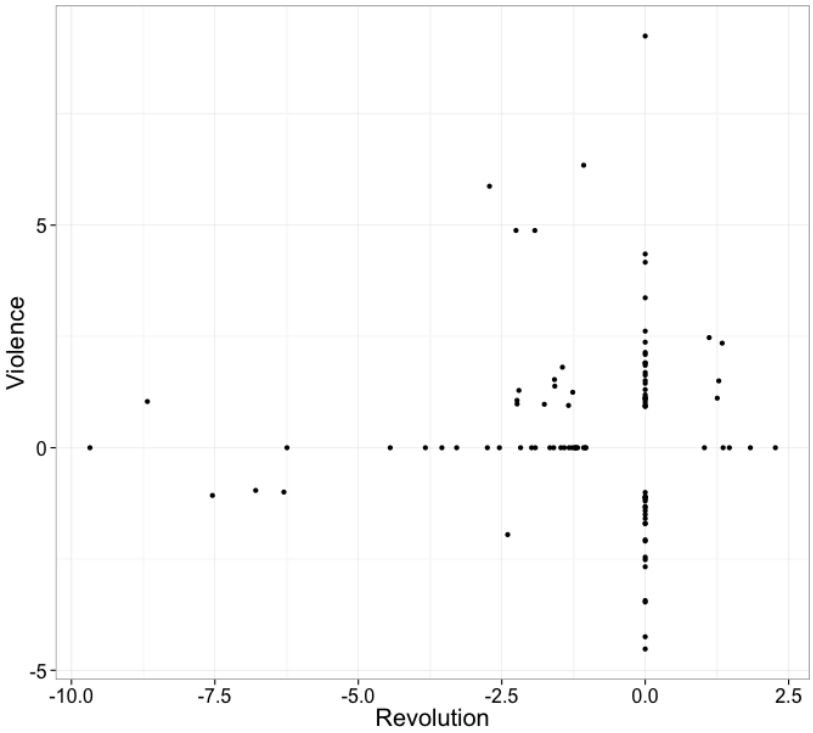

Figure 5 – A scatterplot of the valences of all topics during January of 2011 along the revolution (x-axis) and violence (y-axis) beliefs as computed via Equation 2. Topics within two standard deviations of the mean of both beliefs have been removed.

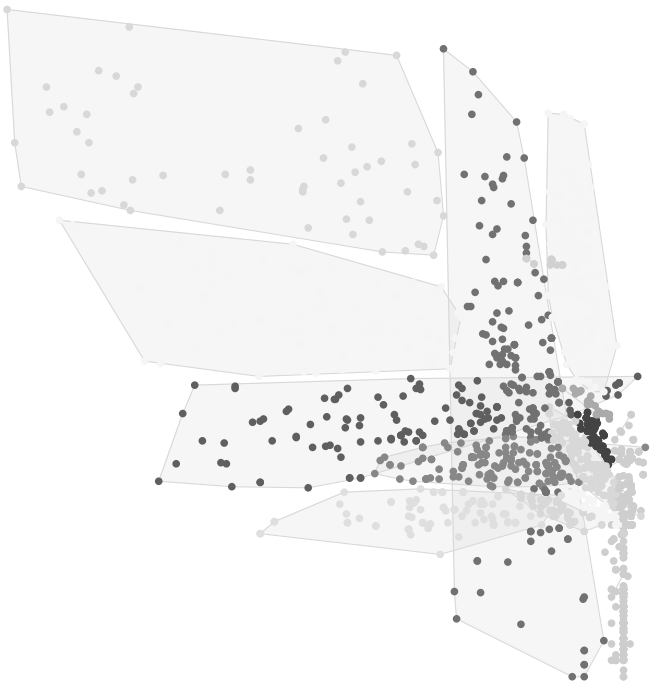

Figure 6- A scatterplot of all agents discovered in the text in January 2011 associated with Egypt. The x-axis of the plot represents the agents' revolution belief and the y-axis represents their violences belief. Agents are greyscaled based on the social group they have been assigned to – because this makes it difficult to discern groups, we also add a bounding polygon around groups.

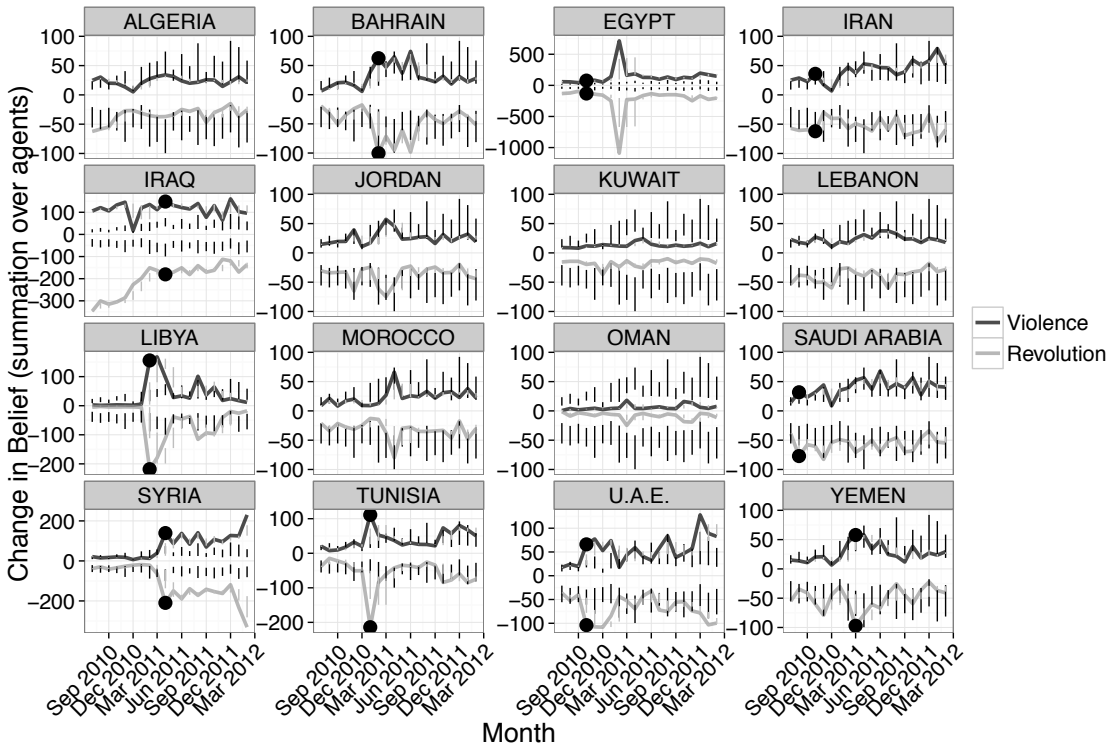

Figure 7- A plot of the sum of the belief change from the beginning until the end of the simulation for agents in each of the sixteen nations studied. On the x-axis, time is represented. Grey and black lines represent change in revolution and violence beliefs over time, respectively. Dark Grey and black vertical bars at each month represent intra-country and inter-country ranges, respectively. Large dots represent the points at which the model predicted revolution- if these dots do not appear for a given country, revolution was never predicted. Note that scales

Tables

	Belief nodeclass	Agent nodeclass	Agent Groups nodeclass	Knowledge nodeclass
Agents	Agent by Belief network	Agent by Agent network	Agent by Group network	Agent by Knowledge network
Knowledge	Knowledge by Belief network			

Table 1 - The necessary node classes and networks to be mined from the raw text

Pro Revolution	**Anti Revolution**	**Pro-violence**	**Anti-violence**
CRIMINAL FALSE IMPRISONMENT	EMBASSIES & CONSULATES	BOMBINGS	ARMS CONTROL & DISARMAMENT
ETHNIC GROUPS	FOREIGN RELATIONS	DEATH & DYING	ARTISTS & PERFORMERS
ECONOMIC CRISIS	HEADS OF STATE & GOVERNMENT	FIREARMS	CHARITIES
ELECTION FRAUD	INTERNATIONAL RELATIONS	GENOCIDE	COLLEGES & UNIVERSITIES
HUMAN RIGHTS VIOLATIONS	SOCIAL JUSTICE	SUICIDE BOMBINGS	ENTERTAINMENT & ARTS
POLITICAL CORRUPTION	US PRESIDENTS	TERRORISM	PEACE PROCESS
PROTESTS & DEMONSTRATIONS	INTERGOVERNMENTAL TALKS	TERRORIST ORGANIZATIONS	PEACEKEEPING
RIOTS	TALKS & MEETINGS	TORTURE	SPORTS
MILITARY OPERATIONS	TRADE DEVELOPMENT	CRIMES AGAINST HUMANITY	UNITED NATIONS INSTITUTIONS
BOYCOTTS	TAX TREATIES & AGREEMENTS	WAR & CONFLICT	WEAPONS DECOMMISSIONING

Table 2- The list of "seed topics" used

Country	Model Prediction	Date of Government Overthrow
ALGERIA	None	None
JORDAN	None	None
KUWAIT	None	None
LEBANON	None	None
MOROCCO	None	None
OMAN	None	None
EGYPT	10/10	2/11
TUNISIA	1/11	1/11
LIBYA	2/11	8/11
BAHRAIN	2/11	None (Revolution squashed around 3/11)
YEMEN	3/11	1/11
SYRIA	4/11	None (Civil War began around 3/11)
SAUDI ARABIA	8/10	None
IRAN	10/10	None
U.A.E.	10/10	None
IRAQ	4/11	None

Table 3- Model prediction results. Predictions were correct for rows colored white (top rows of the table), inconclusive for rows colored light grey (middle), and incorrect for rows colored dark grey (bottom). Within each color group, countries are sorted by the month in which the model predicted revolution.