Applying Cognitive Fusion to Space Situational Awareness

Steven Ingram
Lockheed Martin
Space Systems Company
Sunnyvale, CA 94089
Email: steven.ingram@lmco.com

Margaret Shaw
Lockheed Martin
Space Systems Company
Sunnyvale, CA 94089
Email: margaret.m.shaw@lmco.com

Moses Chan
Lockheed Martin
Space Systems Company
Sunnyvale, CA 94089
Email: moses.w.chan@lmco.com

Abstract

With recent increases in capability and frequency of rocket launches from countries across the world, maintaining a state-of-the-art Space Situational Awareness model is all the more necessary. We propose a fusion of real-time, natural language processing capability provided by IBM cognitive services with ground-based sensor data of positions and trajectories of satellites in all earth orbits. We believe such insight provided by cognitive services could help determine context to missile launches, help predict when a satellite of interest could be in danger, either by accident or by intent, and could alert interested parties to the perceived threat.

We seek to implement an improved Space Situational Awareness model by developing a dynamic factor graph model informed by the fusion of ground-based ”structured” sensor data with ”unstructured” data from the public domain, such as news articles, blogs, and social media, in real time. To this end, we employ IBM’s Cognitive services, specifically, Watson Discovery. Watson Discovery allows real-time natural language processing of text including entity extraction, keyword search, taxonomy classification, concept tagging, relation extraction, sentiment analysis, and emotion analysis.

We present various scenarios that demonstrate the utility of this new Space Situational Awareness model, each of which combine past structured information with related open source data. We demonstrate that should the model come to estimate a satellite is ”of interest”, it will indicate it as so, based on the most pertinent data, such as a reading from a sensor or by information available online. We present and discuss the most recent iterations of the model for satellites currently available on Space-Track.org.

I. Introduction

Despite the public availability of space object observations (e.g., Two Line Elements (TLEs) from space-track.org), context for satellite movements is hard to come by. TLEs are derived from a set of catalog systems whose sources includes input from radar, telescopes, cameras, and other sensors. Traditionally, these sensors are maintained and monitored by a mixture of commercial, academic, and government providers. As numerous as these data providers are–in the tens of thousands–space-track.org can have as few as 10 TLEs an hour of the nearly 19,000 objects being actively tracked in Earths orbit [1], [2]. More so, at the time these TLEs are available, additional information about the reliability of the TLE produced or its likelihood is not published. Information is not available to characterize the satellite, to discern how that TLE was produced, or to estimate how reliable the TLE update is. Space Situational Awareness (SSA) tools generally pull quantitative information (”structured data”) with any textual or natural language information (”unstructured data”) processed and integrated manually. We have constructed the Space Event Risk Assessment (SERA) engine to capture subject matter expert processes and largely quantitative evidence to automate threat assessment and on-orbit event prediction. The tool is meant to monitor space-event streams to identify risks to a list of protected satellites. It combines structured information with expert rules to determine if a threat is present where initially, a maneuver or predicted plane conjunction can combine to form a ”plane conjunction threat” message and with additional updates for TLEs of interest, and other factors, SERA will display a ”close proximity” message. SERA also provides an unweighted tree visualization displaying evidence of identified risks as well 3D visualization or orbits and satellites of interest for a given scenario. An example of a threat tree is shown in Figure 1. More recent events are displayed at the top of the graph while events from farther back in time are at the bottom. Direction of influence is noted with arrows from one node to the next. Maneuvers and predicted plane conjunctions are marked with green circles which when combined can lead to plane conjunction threats, marked as yellow triangles. Plane conjunction threats can combine with maneuvers, predicted plane conjunctions, proximity estimates to form a Close Proximity node, indicated with a red square. Note that nodes in the evidence tree do not have associated probabilities, such as how likely the systems estimation of a predicted plane conjunction is.

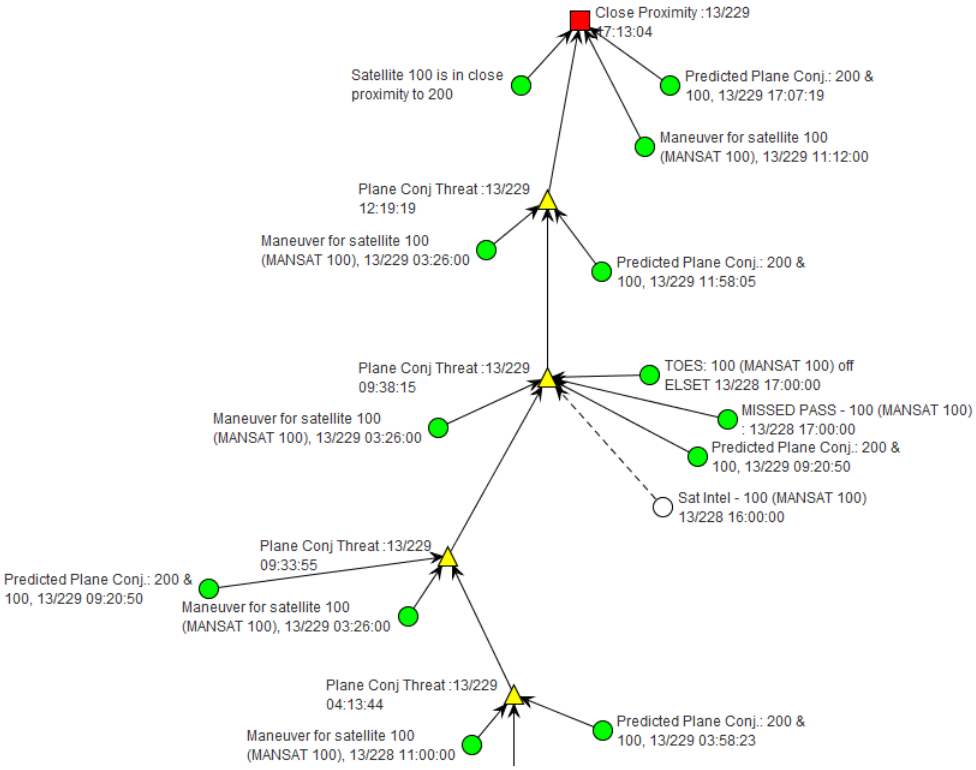

Fig. 1: Example of Evidence Indicating Close Proximity Displayed by SERA as a Tree.

SSA and risk assessment tools like SERA could be improved with the inclusion of probabilistic modeling and fusion of structured and unstructured data. There is a lack of data fusion or formalism where unstructured data (data from open sources published in the news, articles, or social media feeds) complements or further informs the structured data from sensors. Currently SSA software tools used for analysis and tracking of objects in orbit do not automatically make use of or display pertinent information and additional context that is available to the public online, which could provide further insight into the situation. Unstructured information typically must be parsed into computer-digestible formats, removing the ability to perform effective machine learning (e.g., sentiment analysis) on information from a large set of sources. The following three scenarios are presented to illustrate how combining structured and unstructured data sources could provide more context and meaningful information to an observer or operator focused on SSA.

A. Scenario 1

The first scenario is illustrated in Figure 2 and shows relevant events related to the Cosmos 2251 (01670) and Iridium 33 (00633) collision [4]. A few questions that would provide more context to a situation like this include:

- What is the particular satellite's purpose?
- What is the maneuver history of both satellites prior to intercept?
- Why did the satellite maneuver in each case?

Insight into these questions could be derived from public domain data. For example, a Natural Language Processing (NLP) tool such as Watson Discovery could investigate sources to extract related entity mentions with the given satellite and provide links to relevant sources. These references could be combined with SERA capability, which could be leveraged to characterize a maneuver as a risk or if the satellite was only performing an expected maneuver (e.g., for known mission purposes or for station-keeping).

Question	Watson action	SERA action
What is this particular satellite's purpose?	-Fixed options: Commercial, Military, Civil, Science, Exploration, etc. -Guess based on frequency of entity types in articles	
What is maneuver history of both satellites prior to intercept?		Characterize maneuver as risk and provide advanced warning.
Why did the satellite maneuver in each case?	What was the space weather environment like at the maneuver times? – check http://spaceweather.com/	Were objects in the way? Was this a station-keeping maneuver?

Fig. 2: SSA Scenario One.

B. Scenario 2

The second scenario is illustrated in Figure 3 and shows events related to the unexpected launch of TK-1. On March 3, 2017 there was a KT-2 rocket launch from the Jiuquan Satellite Launch Center that took a number of observers by surprise [4]. Again, this scenario could be better informed with analysis from both Watson and SERA. Questions that could be answered in this scenario include:

- What is the recently launched object?
- Is it on a collision with a military satellite?
- Did any world leaders make a statement about the launch?

Like the previous scenario, these questions could be better answered by combining Watson's capabilities for NLP with SERA's risk assessment ability. Watson Discovery could aggregate co-mentions of the satellite of interest with mentions of missiles, rockets, and more from open source data. SERA can be used to display results of possible forward propagated orbits and their associated collision risk, now with the added computer-digestible context from Watson.

Question	Watson action	SERA action
What is the recently-launched object?	-Missile, Satellite, Balloon, Rocket, (more?) -Watson entities: **Objects**: Spacecraft, SpaceMission, SpaceAgency, Rocket, RocketEngine, RocketEngineFuel, RocketFuel, RocketFunction, RocketManufacturer, BipropellantRocketEngine **Entity Relations**: Event, EventDemonstration, Vehicle, Weapon -Count of number of references to these based on location described in article	
Is it on a collision course with the military satellite over the next 10 days?		Propagate orbits forward and compute collision risk
Did any world leaders make a statement about this launch?	-Simple entity query based on finite list of people -Could use Watson Studio to train if necessary	

Fig. 3: SSA Scenario Two.

C. Scenario 3

A third scenario is shown in Figure 4 and illustrates a collection of Russian proximity operations in orbit. In this scenario, operations occurred at numerous epochs that have been reviewed and analyzed since [4]. As these operations occurred, much about the situation was unclear and raised concern. There was online chatter about the developing on-orbit situation including hypotheses from amateur astronomers and more. Situations like this would benefit from combining information from SERA and Watson. Multiple queries could be used to reference multiple payloads where Watson could return information about any mentions of the satellite's functional purpose or associated rocket, as well as any mentions by world leaders or co-mentions of related entities. Again, this would supplement the information processed and displayed by SERA that could provide an operator with a fuller picture as a scenario unfolds.

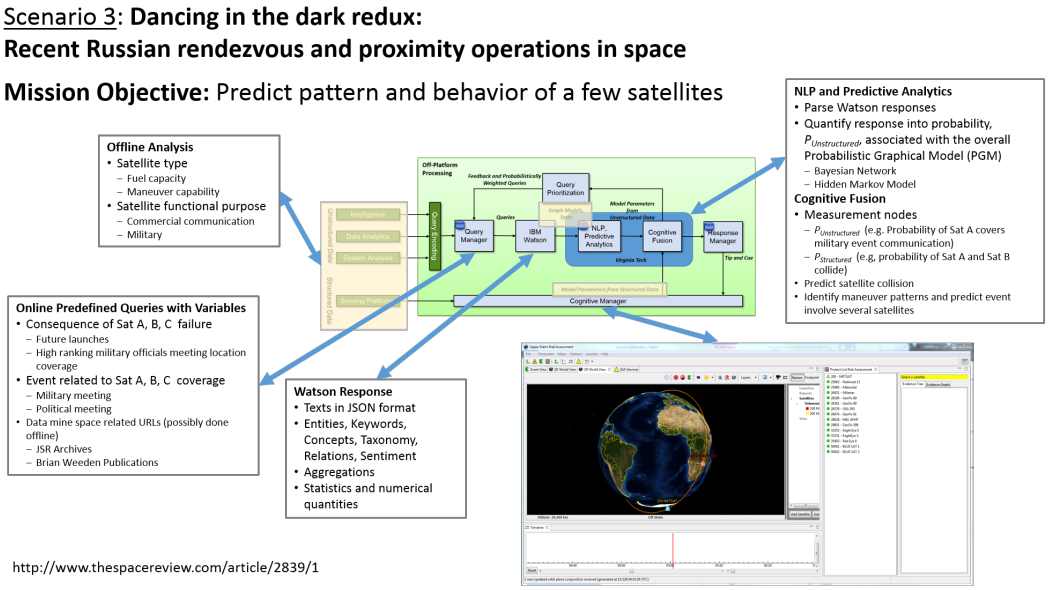

Fig. 4: SSA Scenario Three.

We present our use of DeepDive (Stanfords open source tool) and IBMs cloud computing services (specifically, Watson Discovery) to address the problem of a lack of updates to data for SSA as well as the problem of fusing structured and unstructured data. DeepDive can provide the underlying probabilistic graph model for unstructured and structured fusion, while Watson Discovery can provide NLP markup on unstructured data sources. If formulated appropriately, these tools could work together in real time to continuously update information on every satellite currently being tracked by space-track.org, significantly more than just a handful of satellites at a time [1]. The following section formalizes the problem and our solution.

II. Problem

Fundamentally, the underlying issue with SSA is that much is based only on a rules-based system with no information about relative confidence or estimation on how informative a TLE is. In addition, while orbit propagation is well understood and can be used to forecast a collision days in advance, estimate errors could lead to false alarms.

Figure 5 shows one way SSA could be viewed as a probabilistic graph model where inferencing can be used to guess how likely a satellite is a ”satellite of interest” or not. This is Bayesian network, which is a directed, acyclic graphical model, where arrows between random variables indicate conditional dependencies. Figure 5 illustrates the issue with attempting to reform SSA as a Bayesian Network.

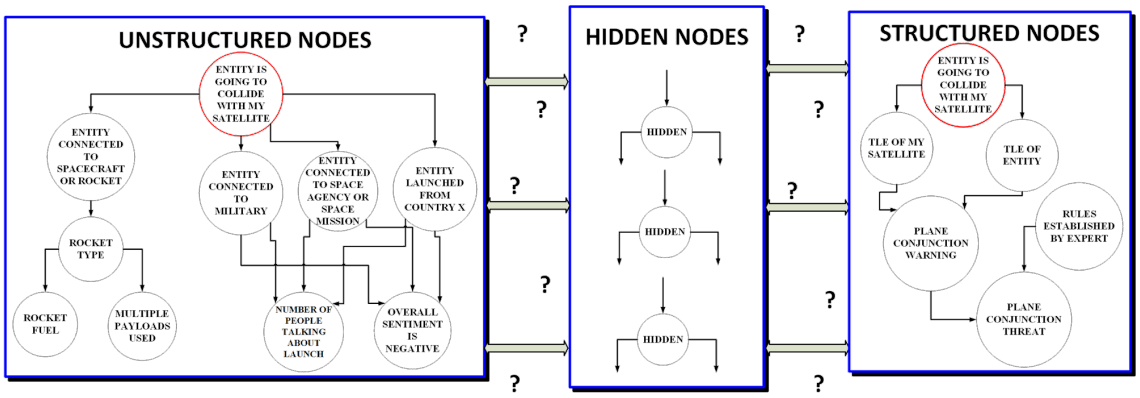

Fig. 5: SSA Bayesian Issue With Combining Unstructured and Structured Data.

While certain relationships and parameters within the SSA domain can be accurately modeled, there is a fundamental problem with combining structured and unstructured data. There are likely significant features unaccounted for and there is not a straightforward way in which to properly weight how each evidence variable might influence the value of query variable.

Classically, graphical models can be used to represent a joint probability distribution $p(\mathbf{y}, \mathbf{x})$, where $\mathbf{y}$ can represent the attributes of entities we want to predict, such as if a satellite is of public interest. The input variables $\mathbf{x}$ represent observed knowledge about the entities such as number of articles mentioning a specific satellite. Unfortunately, modeling the joint distribution can lead to difficulties when using local features that can occur in relational data, because it necessarily requires modeling the distribution $p(\mathbf{x})$, which can include many complex dependencies. Accurately modeling these input dependencies can lead to intractable models, but ignoring them can lead to reduced performance [9]. For example, how would one realiably calculate the probability that a satellite would be mentioned in sociel media in the same paragraph as a specific country? Why not just take the fact that the satellite and a specific country are given instead? Therefore, an alternative solution to this problem is to instead directly model the conditional distribution $p(\mathbf{y}|\mathbf{x})$, which is sufficient for purposes of classification. This is the approach taken by *conditional random fields* [5]. A conditional random field is a conditional distribution $p(\mathbf{y}|\mathbf{x})$ with a corresponding graphical structure. Dependencies among the input variables $\mathbf{x}$ do not need to be explicitly represented because the model is conditional, allowing the potential use of global features of the observed variables.

Formally, we will use data from multiple sources to form an uncertain database, which can be viewed as a set of possible worlds W. Each $\omega \in W$ is an instance of the uncertain database. We define W with a probability distribution $\eta : W \rightarrow [0, 1]\ s.t.\ \Sigma_{\omega \in W}\ \eta(\omega) = 1$, providing a distribution over possible worlds. In this approach, η is represented by a factor graph, which can encode Bayesian and Markov networks [7]. A factor graph can easily represent relationships between random variables with complex dependencies, making them an appropriate choice for relational data, specifically data in the SSA domain.

A factor graph, parameterized by θ, is a bipartite graph where nodes consist of the pair $G_\theta = < V, \Psi >$ with $V = X \cup Y$ is the set of random variables where X is the set of observed variables, Y is the set of hidden variables, and $\Psi = \{\psi_k\}$ is the set of factors.

In our case, the random variables represent the range of values that an uncertain element in the database may acquire. Similarly, each hidden variable $Y_i \in Y$ is associated with a domain $D(Y_i)$, representing the range of possible values for each Y_i. Generally, the domain could be binary $\{0,\ 1\}$, enumerations $\{$small,medium,large$\}$ or real-valued $\{r \in \Re\}$. Observed variables are fixed to specific values in the domain and can be considered constant [8]. We let capital letters with a subscript (Y_i, X_i) represent a single random variable, and lowercase letters (y_i, x_i) represent a value from the corresponding domain: $y_i \in D(Y_i)$ We use $X_i = x_i$ to indicate X_i takes on the value x_i. In addition, superscripts are used to denote sets where $X^r = x^r$ means the set of variables $\{X_i, X_{i+1}, \ldots, X_{i+r}\}$ take on values $\{X_i = x_i, X_{i+1} = x_{i+1}, \ldots, X_{i+r} = x_{r+1}\}$ with the assumption that x_i is a value for X_i's domain. A capital letter without a subscripts refers to its entire variable space. Here, Y is all hidden variables and X is all observable variables.

Again, factors model relationships between random variables and even multiple factors can be applied to the same variable by expressing a different function with a different weight that prefers certain (possibly opposite) assignments to that variable [8]. This flexible structure is well suited for modeling real world relational data. Each factor ψ: $x^m \times y^n\ \rightarrow\ \Re_+$ maps assignments to subsets of the observed variables $x^m \subseteq D(X)$ and hidden variables $y^n \subseteq D(Y)$ to a real-valued non-negative value.

In general, the factors are computed as a log-linear combination of a sufficient statistic ϕ_k and corresponding weight $\theta_k\quad as\quad \psi_k(x^m, y^n) = exp(\phi_k(x^m, y^n) \cdot \theta_k)$, with ϕ as the user-specified features used to represent underlying data and with θ as the corresponding real-valued weights measuring feature influence. There are numerous methods available from machine learning and statistics for automatically determining these weights [9]. DeepDive, which will be described later, employs Gibbs sampling for learning the weights.

Given the above formalism, the factor graph G_θ expresses a probability distribution (parameterized by θ and conditioned on X) $\eta_G : X \times Y\ \ \rightarrow\ \ s.t.\ \Sigma_{y \in D(Y)}\ \eta_G(y|x) = 1$. If the graph can be decomposed into a set of factors Ψ (where each $\psi \in \Psi$ has arity $a + b$) then the probability distribution η_G is given as:

$$\eta_G(Y = y|X = x; \theta) = \frac{1}{N_X} \prod_{\psi \in \Psi} \psi(y^a, x^b) \tag{1}$$

where $N_X = \sum_{y \in Y} \prod_{k=1}^{n} \psi_k(y^a, x^b)$ is a normalizing constant ensuring that the distribution sums to 1.

For a possible world, an uncertain database *DB* is a set of relations $R = \{R_i\}$ each with schema S_i^k containing attributes $R_i \cdot a_1, \ldots, R_i \cdot a_k$. A deterministic tuple t for a relation R_i is a realization of a value for each attribute $t = \langle v_1, \ldots, v_k \rangle$ for constants $v_1 \in D(a_1) \ldots v_k \in D(a_k)$. If T is the set of all these tuples for relations in the database then the set of all worlds realizable by this database is $W_D = \{w|w \subseteq T\}$.

Allow every field in the database to be a random variable whose domain is the same as the field attribute's domain and let an observed variable X be a deterministic field, and a hidden variable Y be an uncertain field. The hypothesis space, then, of the random variables (X and Y) contain the set of possible worlds because each field has domain equivalent to its respective attribute. Deterministic factors are able to model constraints over sets of variables by outputting 1 if the constraint is satisfied, and 0 if it is violated. With this in mind, we define W to be all the possible worlds with respect to the factor graph's probability distribution η:

$$W = \{\omega \in W_{DB} | \eta_G(\omega)\} \tag{2}$$

Solving for the world that is most likely given the observed variables is described later in the following section.

III. Approach

We address the problem defined in the previous section by implementing cognitive fusion using DeepDive and IBM's Watson Discovery service.

A. Cognitive Fusion

We seek to deal with the lack of fusion of unstructured and structured data by combining information from sensors that provide structured data as Two-line Element sets, with Natural Language Processing from IBM's Watson as illustrated in Figure 6. Note this is the underlying structure mentioned earlier on which each scenario is based.

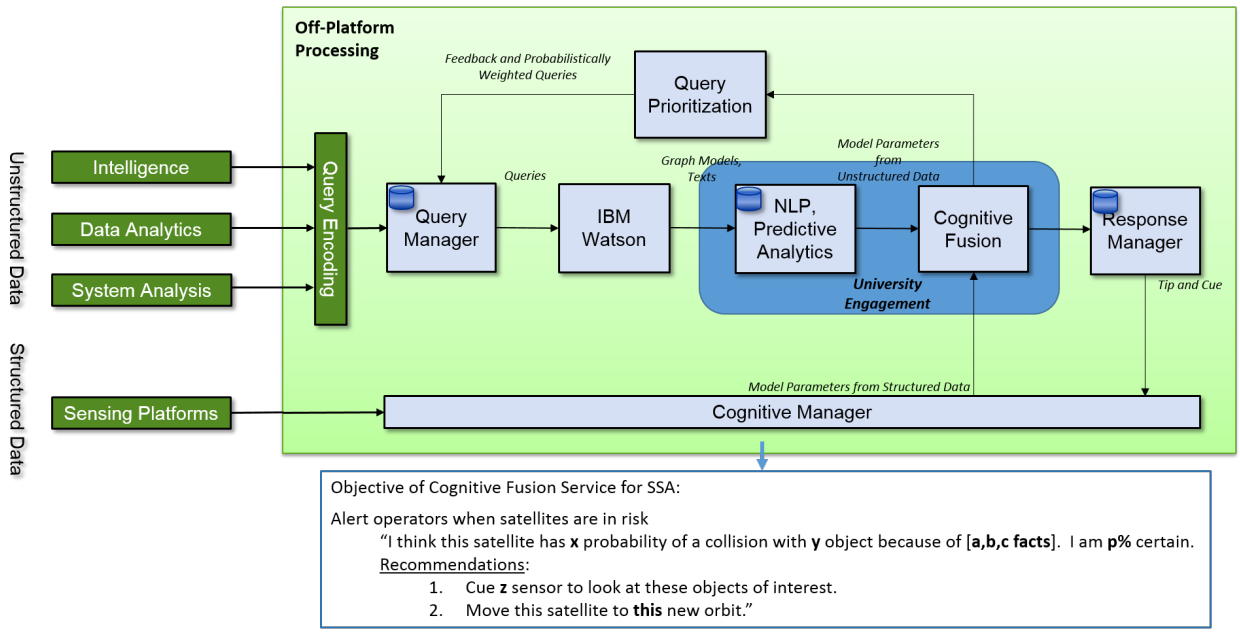

Fig. 6: SSA Cognitive Fusion.

Overall, the system is built to estimate the likelihood a satellite is "of interest" and to suggest to an operator what action, if any, should be taken. Example actions include cuing sensors to further investigate an object or to make preparations to move a satellite to a safe orbit. This estimation is calculated with off-platform processing that takes as input intelligence data, data analytics information, system analysis, and data from sensing platforms as unstructured data. The unstructured data can be loaded into a database after being processed by IBM's Watson Discovery, which can be queried with a REpresentational State Transfer (REST) API call. Simply, requests for information can be made with an HTML request with an embedded query. The result of queries combined with TLEs are combined with Stanford's DeepDive to create a factor graph that ultimately informs the response manager on additional actions. This response manager controls additional actions by the system, as well indicating to the operator satellite status and if maneuver action should be taken. For this section, we primarily focus on IBM Watson, NLP, Predictive Analytics, and Cognitive Fusion information blocks illustrated in Figure 6.

We seek to improve overall SSA scenario by continuously updating the likelihood that a satellite is considered "of interest" for every satellite track published by space-track.org. This likelihood will be estimated with a factor graph that combines information from TLEs and from unstructured sources, like news articles.

B. DeepDive

This framework of instantiating a statistical model for each satellite informed by each satellite's TLE and the most recent, relevant article markup determined by IBM's Watson will use DeepDive. DeepDive creates structured data (SQL tables) from unstructured information (text documents) and integrates such data with an existing structured database. DeepDive is used to extract sophisticated relationships between entities and make inferences about facts involving those entities. DeepDive is a type of data management system that enables one to tackle extraction, integration, and prediction problems in a single system,

which allows users to rapidly construct sophisticated end-to-end data pipelines, such as dark data BI (Business Intelligence) systems. By allowing users to build their system end-to-end, DeepDive allows users to focus on the portion of their system that most directly improves the quality of their application [3]. Example random variables that we could query or make an inference on include whether a satellite just performed a routine maneuver or not, if the satellite is close to another or not, whether the satellite is of public interest or not, and finally, if the satellite is "of significant interest" or not. "Of interest" is a broad term which could describe a satellite behaving in an uncharacteristic or an anomalous fashion. The random variables that will be used as evidence will be the TLE for each satellite and the list of entities associated with each satellite based on the article more relevant to the satellite. These random variables are related by weighted factors. Weights based on NLP markup from Watson are learned by the system and weights from TLE information are set manually. These four factors are all implication functions. In logic, this simply means that $f(A, B)$ is equivalent to "A implies B" and can be written as $A \rightarrow B$. The logical implication means if A is TRUE, then we expect B to also be TRUE. In addition, if A is FALSE, then we cannot necessarily expect B to also be FALSE. For each satellite, it's TLE, based on certain parameters, can imply that a satellite has performed a routine maneuver. Information from a satellite's TLE can also imply it is close to another satellite. The NLP markup from Watson can imply that a satellite is of public interest. An example diagram is shown in Figure 7.

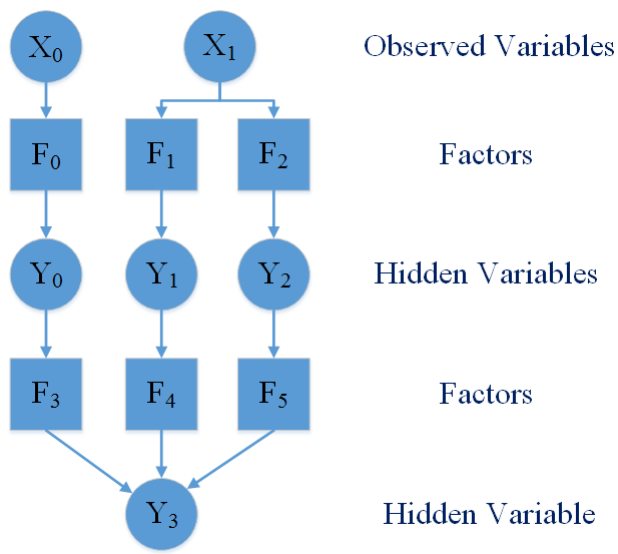

Fig. 7: Example Factor Graph for a Given Satellite.

Formally, as an exmaple for each satellite in our model, we have $F_0(X_0, Y_0)$, $F_1(X_1, Y_1)$, $F_2(X_1, Y_2)$. With this, we can say a TLE with certain characteristics imply a satellite has performed a routine maneuver and is close to another satellite, and that NLP markup implies the satellite is of public interest. For the next part of the model for each satellite we define $F_3(Y_0, Y_3)$, $F_4(Y_1, Y_3)$, $F_5(Y_2, Y_3)$. These three implications functions mean that a satellite being of public, not performing a routine maneuver, or being close to another satellite implies that satellite is "of interest." Similarly, a satellite, based on TLE information could be marked as debris or decayed and would therefore imply it is not a satellite of interest.

1) Solving the Factor Graph: Each factor function has a corresponding weight, which describes how much influence the factor has on its variables in relative terms. Basically, the weight encodes the confidence in the relationship expressed by the factor function. If the weight is high and negative, we are more confident that the function is incorrect; if the weight is high and positive, we are confident in the function the factor encodes. With DeepDive, these weights can be learned from training data or assigned manually based on domain-specific knowledge [3].

Recall from Section II, a possible world is one with an assignment to every variable in a factor graph. Generally, the possible worlds are not equiprobable, and it more likely that each possible world has a different probability of existing. The probability of a possible world is directly proportional to the weighted combination of all the factor functions in the graph, evaluated at the values specified by the possible world. For weights to be learned, training data is required, which will define a set of possible worlds where DeepDive's learning process will choose weights by maximizing the probabilities of the possible worlds.

Due to the intractable problem of exact inference on factor graphs, an approximate alternative commonly used is Gibbs sampling. The benefit of Gibbs sampling is that given a multivariate distribution, it is much less computationally expensive to sample from a conditional distribution than it is to marginalize by integrating over a joint distribution. Simply, this process begins from a random possible world and iterates over each variable X, Y, updating its value by evaluating the factor functions of the factors that X, Y is connected to and the values of the random variables connected to those factors. This is known as the

Markov blanket of v. After a sufficient number of iterations of this process over the random variables in the factor graph, we can calculate the number of iterations during which each variable had a specific value and use the ratio between this quantity and total number of iterations to compute an estimate of the probability of the variable taking that value.

Figure 8 shows the factor graph for each satellite in the SSA DeepDive database.

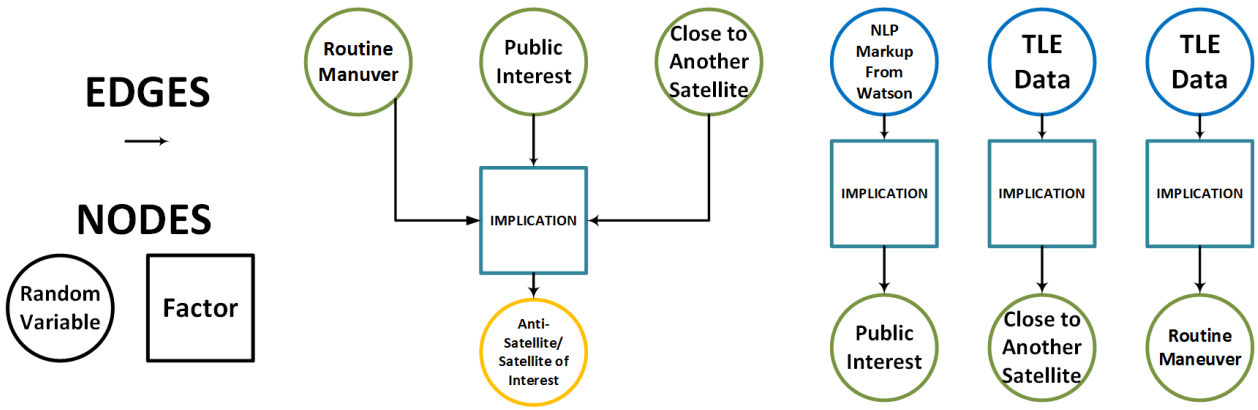

Fig. 8: SSA DeepDive Factor Graph For Each Satellite.

2) Gibbs Sampling Procedure: To perform learning and inference on the factor graph with DeepDive, we employ the following Gibbs sampling procedure [10]. Suppose we will obtain $\mathbf{k}$ samples of $\mathbf{W} = (\mathbf{w}_1, \ldots, \mathbf{w}_n)$ from a joint distribution $\mathbf{p}(\mathbf{w}_1, \ldots, \mathbf{w}_n)$. Denoting the ith sample by $\mathbf{W}$. The process is as follows:

1) Start with an initial value $\mathbf{W}^{(i)}$
2) Obtain next sample, denoted $\mathbf{W}^{(i+1)}$, where $\mathbf{W}^{(i+1)} = (\mathbf{w}_1^{(i+1)}, \ldots, \mathbf{w}_n^{(i+1)})$ is a vector. We sample each component of the vector, $\mathbf{w}_j^{(i+1)}$, from the distribution of that component conditioned on all the other components that have been sampled so far. Note, we condition on $\mathbf{W}^{(i+1)}$'s components up to $\mathbf{w}_{j-1}^{(i+1)}$, and thereafter condition on $\mathbf{W}^{(i)}$'s components, starting from to . To do this, we sample the components in order, starting from the first component. In short, to sample $\mathbf{w}_{j-1}^{(i+1)}$, we update it according to the distribution specified by $\mathbf{p}(\mathbf{w}_j^{(i+1)} | \mathbf{w}_1^{(i+1)}, \ldots, \mathbf{w}_{j-1}^{(i+1)}, \mathbf{w}_{j+1}^{(i)}, \ldots, \mathbf{w}_n^{(i)})$. Note the value that the $j+1th$ component had in the ith sample is used, not the $i+1th$ sample.
3) Repeat the previous steps $\mathbf{k}$ times.

This sampling processes approximate the joint distribution of all variables. Second, the marginal distribution of any subset of the variables can approximated by only considering the samples for that subset of variables, the rest can be ignored. Finally, the expected value of any variable is approximately the average over all the samples.

C. Watson

IBM offers a suite of cloud computing services that can be accessed via their Bluemix console on a web browser, or generally, through REST API calls. One in particular is the IBM Watson Discovery service. Its features include automated document ingestion, conversion, natural language processing enrichment, and a large collection of pre-enriched news content updated daily. Figure 9 shows NLP enrichments provided by Watson Discovery and provides a short description of each. Enrichments include entity extraction, keyword extraction, taxonomy classification, concept tagging, relation extraction, sentiment analysis, and emotion analysis [6]. This collection of articles marked up with natural language processing enrichments are made available for fast querying and aggregation. For demonstration purposes of utility for SSA, we will use the entity extraction and keyword extraction feature to create weights for factors for DeepDive to learn. Anytime a new TLE is pulled from Space-track.org, Watson Discovery is queried with the international satellite designator from the TLE and used as the search term. The top article, scored by relevancy (to the search query entered) by Watson, will be used for entity and keyword extraction. Every entity and keyword in that article is stored, counted, scored, and classified by Watson Discovery. That information is concatenated into separate variable-length strings such that all entities and keywords and their attributes form one collection of processed unstructured data per satellite.

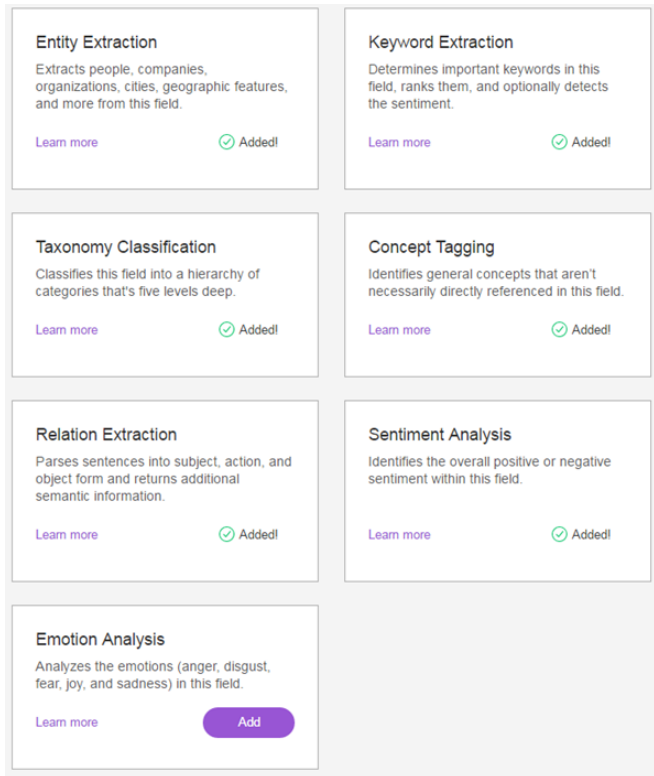

Fig. 9: Watson Discovery Enrichments.

IV. Results

Our initial results confirm our ability to monitor all satellite tracks published by space-track.org in real-time, while simultaneously considering unstructured and structured data. The following set of figures show data pulled from space-track.org, n2yo.com, and NLP markup from Watson Discovery.

Figure 10 shows the HTML response from space-track.org when an API call is made to request Two-Line Element Sets (TLEs) of satellites. The TLE consists of a number of values that help identify the satellite as well describe its orbit at a specific epoch. Some of these values include Inclination, Right Ascension, Eccentricity, and more. For the purposes of this project, we use the Name of the Satellite and it's International Designator for open source searches. Each of these two-line elements are stored as a one-line string into a table where the NORAD ID of the satellite is a key. The international designator of the satellite is later extracted to collect additional information from n2yo.com and is also used to query against Watson.

```
1 42915U 17047A   17230.66484481 -.00000375  00000-0  00000+0 0  9999
2 42915  26.2144 331.4020 5851760 179.9686  95.5371  2.00361233    03
1 42916U 17047B   17232.22896835 -.00000235  00000-0  00000+0 0  9999
2 42916  26.4397 330.7189 5826374 179.2077 182.5375  2.06615587    38
1 42917U 17048A   17232.33580976 -.00000022  00000-0  00000+0 0  9993
2 42917  19.9475  87.8654 7237249 179.1490 183.6767  2.30515890    26
1 42918U 17048B   17231.80813103 -.00000019  00000-0  00000+0 0  9992
2 42918  19.9600  88.1050 7248448 179.3057  97.2832  2.26329709    03
```

Fig. 10: Response from Space-Track API Call for TLE Data.

Figure 11 shows the relevant portion of n2yo.com that is used to collect additional open source information about each satellite. The site doesn't have an API for web-scraping purposes, but data for each satellite can be collected easily when the NORAD ID is used as part of the web address in HTML requests [2]. Data from this site include launch date of the satellite, launch site, satellite common name, and more. This information is stored in a table as a single string into a table where the NORAD ID is used as a key.

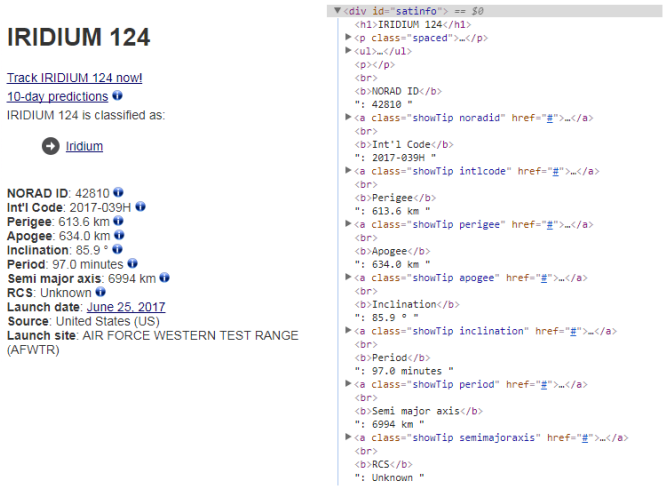

Fig. 11: HTML From N2YO Site.

Figure 12 shows a screenshot of the unprocessed JSON output from Watson. This file is parsed such that only the information about the entities and keywords from the most relevant article associated with the satellite is stored. The entity and keyword information is collected as two separate strings. Those strings are then stored in separate tables where the key is the NORAD ID of the satellite. This produces two of the random variables that implies whether a satellite is of public interest.

Fig. 12: JSON Output from Query to Watson Discovery.

To summarize the overall quality of DeepDive results, calibration plots are generated after each learning process. Each random variable is assigned a marginal probability because DeepDive uses a joint probability model. Basically, if one considers all the ”facts” which DeepDive assigns a probability score of 0.90, then 90% of these facts are correct. DeepDive programs define one or more prediction and test sets for each relation, which is a collection of labeled data for that particular relation. This set is used by DeepDive to create a calibration plot for each random variable. Figure 13 shows an example calibration plot for the ”satellite of interest” relation for the general SSA application, which provides an aggregated view of how the system behaves.

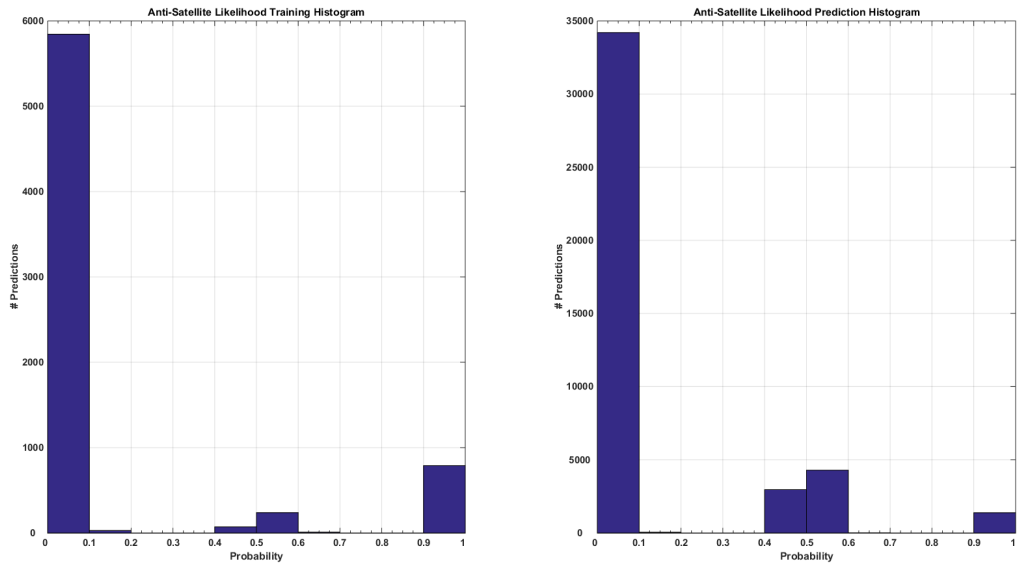

Fig. 13: Results Calibration Plots For ”Satellites of Interest”.

As shown in Figure 13 the calibration plot show two important figures:(a) # prediction (test set), which measures the number of extractions in the test with a certain probability, and (b) # predictions (whole set), which measures the number of extractions in the whole set with a a certain probability. The total number of satellites in the first graph is equal to 5979. This graph includes labeled data used to train the model as well as labeled data held out for the model to make predictions on. The second graph displays results for 42895 satellites. This value is all inclusive, it displays what the model predicted for all labeled and unlabeled data. For these plots, probabilities are placed into bins as a histogram. The test set has labels that were provided previously so we can measure accuracy, while the whole set does not. We create a histogram of the number of predictions in each bin. In a well-tuned system, this plot should have something similar to a ”U” shape. We want most of the extractions concentrated at high probability and low probability bins. Figure 13 shows a mostly U-shaped curve with some masses in the 0.4-0.5 bin and the 0.5-0.6 bin. This would suggest there are some satellites for which the system has insufficient features and the system is unable to determine if the extraction is true or false. Extractions that fall into all the bins not in (0,0.1) or (0.9,1.0) are candidates for improvements and may require additional features to push these probabilities into either (0,0.1) or (0.9,1.0). The final histogram in the figure illustrates the behavior for which we do not necessarily have training examples generated by supervision or human labeling. Visually, this histogram should have a similar shape to (a), albeit with more extractions. If the shapes are dissimilar that would suggest some bias in the selection of the hold-out set or overfitting to the labeled data. To quantify the performance of our model, we use a confusion matrix to measure accuracy, misclassification rate, true positive rate, false positive rate, and specificity. Figure 14 shows the confusion matrix for our model that uses data from space-track.org, N2YO, and entity markup from Watson. Figure 15 shows the accompanying performance metrics.

n=1732	**Predicted: FALSE**	**Predicted: TRUE**	
Actual: FALSE	TN = 1450	FP = 45	1495
Actual: TRUE	FN = 53	TP = 184	237
	1503	229	

Fig. 14: Confusion Matrix Using Structured and Unstructured Data

METRIC	**VALUE**
ACCURACY	0.94
MISCLASSIFCATION RATE	0.06
SENSITIVITY / RECALL	0.78
FALSE POSITIVE RATE	0.03
SPECIFICITY	0.97

Fig. 15: Metrics Using Structured and Unstructured Data

From the confusion matrix we can compute the Accuracy of our model or how often our classifier is correct. We have

$(TP + TN)/total = (1450 + 184)/1732 = 0.94$ or 94% accuracy. Conversely, our misclassification rate, or how often our model is wrong, is simply $(FP + FN)/total = (45 + 53)/1732 = 0.06$ or 6%. We can also compute True Positive Rate, or when a satellite is "of interest," how often our model predicts True. We have $TP/$ Actually True$= 184/237 = 0.78$. Note this is also known as "Sensitivity" or "Recall." We can also compute the False Positive Rate, or when an extraction should be classified as False, how often our model predicts False. From the matrix, we have $FP/$Actually False$= 45/1495 = 0.03$. Further, we can quantify specificity, or when the model should predict False for a satellite of interest, how often it predicts False. We compute this by simply subtracting False Positive Rate from 1 and in this case we have a Specificity value of 0.97. Figure 15 summarizes these values.

To quantify the success of the overall system we also present the output of two preliminary models, one that uses only structured data, and one that uses only unstructured data. Figure 16 and Figure 17 show the confusion matrix for the model built only using unstructured data and the accompanying performance metrics, respectively. Note the accuracy is only 17%.

n=1536	**Predicted: FALSE**	**Predicted: TRUE**	
Actual: FALSE	TN = 47	FP = 1229	1276
Actual: TRUE	FN = 46	TP = 214	260
	93	1443	

Fig. 16: Confusion Matrix Using Only Unstructured Data

METRIC	**VALUE**
ACCURACY	0.17
MISCLASSIFCATION RATE	0.83
SENSITIVITY / RECALL	0.82
FALSE POSITIVE RATE	0.96
SPECIFICITY	0.04

Fig. 17: Metrics Using Only Unstructured Data

Figure 18 and Figure 19 show the confusion matrix for the model built only using structured data and the accompanying performance metrics, respectively. Note the accuracy is only 65%.

n=1488	**Predicted: FALSE**	**Predicted: TRUE**	
Actual: FALSE	TN = 942	FP = 297	1239
Actual: TRUE	FN = 231	TP = 18	249
	1173	315	

Fig. 18: Confusion Matrix Using Only Structured Data

METRIC	**VALUE**
ACCURACY	0.65
MISCLASSIFCATION RATE	0.35
SENSITIVITY / RECALL	0.07
FALSE POSITIVE RATE	0.24
SPECIFICITY	0.76

Fig. 19: Metrics Using Only Structured Data

As these figures show, the model is improved when both structured data and unstructured data sources are incorporated together. The model based on structured data outperforms the model based only on unstructured data, while most metrics, including accuracy, misclassification, false positive rate, and specificity are improved when combining unstructured and structured data. Using structured data alone, the accuracy of the model is only 65%, but when combined with unstructured data is included, the accuracy is improved to 94%. Similarly, misclassification is improved from 35% to 6%, sensitivity is reduced from 0.82 to 0.76, false positive rate is improved from 0.24 to 0.03, and specificity is improved from to 0.76 to 0.97.

We believe the model could be further improved by incorporating more features that would help distinguish satellites that were not properly classified. We have many options going forward, including implementing word vectorization, TLE vectorization, relation extraction, maneuver characterization, and more.

V. Conclusion

In this paper, a cognitive fusion framework for SSA applications making use of Stanford's DeepDive and IBM's Watson Discovery was presented as a novel approach to making real-time assessments of satellites in orbit and determining which would be "of interest" to an operator. This underlying framework combines unstructured data from open sources, like news articles, blog posts, and social media posts with structured data from sensor systems. Example scenarios were given to demonstrate

multiple use cases where combining unstructured data available from public sources with structured data from sensors could be more informative than either set analyzed alone. The problem was defined formally as solving for the hidden parameters and weights for a factor graph via supervision and Gibbs sampling. To realize this, a DeepDive application was developed and the application incorporated TLEs with NLP markup from Watson that included entity and keyword information most associated with each satellite. Results were presented and demonstrated that the majority of satellites had sufficient features describing them and that the system was able to place their likelihood of being ”interesting” into probability bins of high certainty (greater than 0.9 or less than 0.1). With our current model, we achieved an overall Accuracy of 94%. The opportunities for development and refinement are numerous. With the success of the current model, it would be quite feasible to tune the system for other specific cases, as desired by the operator. Additional features could be incorporated and other constraints could be implemented as required.

ACKNOWLEDGMENT

The authors would like to thank David Cappellucci for providing his expert opinion, continued support, and insight into SSA and for providing useful data sources for various SSA scenarios.

REFERENCES

[1] Space-Track.org. (2017). *Space-Track*. [online] Available at: http://space-track.org [Accessed 21 Aug. 2017].

[2] N2YO.com - Real Time Satellite Tracking and Predictions. (2017). *N2YO.com - Real Time Satellite Tracking*. [online] Available at: http://www.n2yo.com/ [Accessed 21 Aug. 2017].

[3] DeepDive.stanford.edu. (2017). *DeepDive*. [online] Available at: http://deepdive.stanford.edu [Accessed 21 Aug. 2017].

[4] Thespacereview.com. (2017). *The Space Review: essays and commentary about the final frontier*. [online] Available at: http://www.thespacereview.com/ [Accessed 21 Aug. 2017].

[5] J. Lafferty, A. McCallum, and F. Pereira. *Conditional random fields: Probabilistic models for segmenting and labeling sequence data*. Proc. 18th International Conf. on Machine Learning, 2001.

[6] IBM.com. (2017). *Discovery - API Reference — IBM Watson Developer Cloud*. [online] Available: https://www.ibm.com/watson/developercloud/discovery/api/v1/n[Accesse 21 Aug. 2017].

[7] DeepDive.stanford.edu (2017). *Factor Graph* [pdf] Available at: http://deepdive.stanford.edu/assets/factor_graph.pdf [Accessed 21 Aug. 2017].

[8] Alex.smola.org. (2017). *Graphical Models*. [online] Available at: http://alex.smola.org/teaching/cmu2013-10-701x/pgm.html [Accessed 21 Aug. 2017].

[9] Sutton, C. (2012). An Introduction to Conditional Random Fields. *Foundations and Trends in Machine Learning*, 4(4), pp.267-373.

[10] C. Zhang and C. Ré, ”Towards High-Throughput Gibbs Sampling at Scale”, *Proceedings of the 2013 International Conference on Management of Data - SIGMOD* ’13, 2013.