Learning a Resource Scale for Collectible Card Games

Gianlucca Zuin
Departamento de Ciência da Computação
Universidade Federal de Minas Gerais
Belo Horizonte, Brazil
gzuin@dcc.ufmg.br

Adriano Veloso
Departamento de Ciência da Computação
Universidade Federal de Minas Gerais
Belo Horizonte, Brazil
adrianov@dcc.ufmg.br

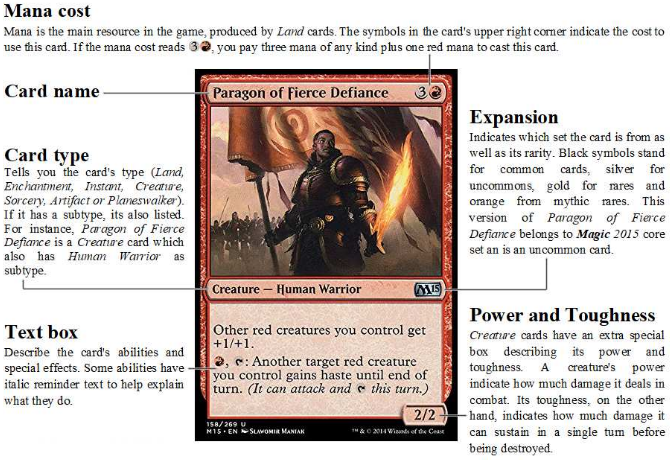
***Abstract*—In Collectible Card Games like "Magic: the Gathering", one of the developers' main challenges is creating new and interesting cards that are not too strong or game-braking, pertaining the game's overall balance. One way to address this issue is through the analysis of the cards resource costs. Powerful cards need more resource to be played while weaker ones need less resource. This work proposes a recommender system to a card's resource scale. In summary, we model the problem as a classification task and present and in-depth analysis of our results. We propose using LSTMs to learn a vector representation for text followed by XGBoost models to incorporate remaining features. Our approach is capable of reaching a Mean Reciprocal Rank of 0.8064 despite superficially identical cards having different mana costs. The analysis provided indicate that the model was able to learn useful rules for predicting a card's resource cost and highlight key insights for future research.**

***Index Terms*—Collectible Card Games, Game Balancing, Deep Learning, Gradient Boosting**

I. Introduction

Collectible Card Games (CCG), also called a *Trading Card Games* (TCG), are games played with specially designed sets of cards. The modern concept of CCG was first presented in "Magic: The Gathering", designed by Richard Garfield and published by *Wizards of the Coast* in 1993 [1]. One of Magic's core cards are lands. Each player can play one land card per turn and these are responsible for resource generation. Each other kind of card has a resource cost, called mana cost, which informs the amount and color of mana needed to play the given card. Figure 1 illustrates the basic features of Magic cards.

During the span of its 26 years of existence, "Magic: the Gathering" published over 17,000 different cards with various mana costs, as illustrated in Figure 2. In Magic and several other CCGs, a way of balancing cards is addressing the amount of resource needed to use them. Powerful cards with various abilities require larger amounts of mana to be spent in order for them to be played, while simpler ones require less mana. However, this balancing is far from perfect. One consequence is the occurrence of *'broken'* and *'useless'* cards that provide way too strong or weak effects respectively in comparison to their mana cost.



Mana cost
Mana is the main resource in the game, produced by *Land* cards. The symbols in the card's upper right corner indicate the cost to use this card. If the mana cost reads 3●, you pay three mana of any kind plus one red mana to cast this card.

Card name

Card type
Tells you the card's type (*Land, Enchantment, Instant, Creature, Sorcery, Artifact or Planeswalker*). If it has a subtype, its also listed. For instance, *Paragon of Fierce Defiance* is a *Creature* card which also has *Human Warrior* as subtype.

Text box
Describe the card's abilities and special effects. Some abilities have italic reminder text to help explain what they do.

Expansion
Indicates which set the card is from as well as its rarity. Black symbols stand for common cards, silver for uncommons, gold for rares and orange from mythic rares. This version of *Paragon of Fierce Defiance* belongs to ***Magic*** *2015* core set an is an uncommon card.

Power and Toughness
Creature cards have an extra special box describing its power and toughness. A creature's power indicate how much damage it deals in combat. Its toughness, on the other hand, indicates how much damage it can sustain in a single turn before being destroyed.

Paragon of Fierce Defiance 3●
Creature — Human Warrior
M15
Other red creatures you control get +1/+1.
●, ↷: Another target red creature you control gains haste until end of turn. *(It can attack and ↷ this turn.)*
2/2
158/269 U
M15 • EN Slawomir Maniak
™ & © 2014 Wizards of the Coast

Fig. 1. Core concepts that encompass a "Magic: The Gathering" card.

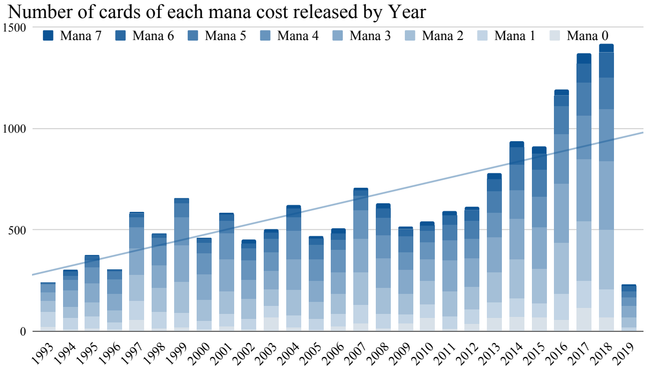


Fig. 2. Number of cards released in each year with mana cost up to 7. Data collected in March of 2019.

Designing for balance is core in competitive games. Ensuring fairness in player vs. player games is crucial to the success of any game that features this sort of interaction. This is a particularly relevant problem in Magic. Cards that are too strong end up being banned or have their monetary value inflated, preventing access to most players. The weak cards, in turn, end up not being chosen to constitute the decks of many players. This phenomenon leads to many cards not being seen

This work was partially funded by authors individual grants from CNPq and FAPEMIG.

978-1-7281-1884-0/19/$31.00 ©2019 IEEE

in actual play, which not only harms the competitive scenario giving players less options, but also consists in the waste of time and effort of the development team.

This work addresses the challenge of proposing balanced mana cost for cards in "Magic: the Gathering" CCG. We propose a resource scale method based on cards already published. We model this task as a learning the multi-class classification problem. We employ a LSTM to quantify the influence of a card's textual patterns in its cost and extract its vector representation from the output of the penultimate layer. We also explore the usage of both Multi-Layer Perceptrons and XGBoost to join this representation with other handcrafted features, some of which are illustrated in Figure 1.

The proposed model is able to learn card's cost abstractions, reaching $+0.8$ MRR (Mean Reciprocal Rank) in the general case and $+0.93$ MRR when it is allowed to abstain from predicting under ambiguous interpretations yet still handling over half the cards in the dataset of existing cards. It is worth noticing that the data could be interpreted as noisy, as there are cards with the same features and abilities but with different mana costs, as well as cards with the same mana cost and features but with distinctly different abilities.

II. Related Work

Although many works feature "Magic: The Gathering", to our knowledge, none attempt to actively tackle the issue of game balancing. In particular, no work addresses the task of suggesting mana costs for new and existing cards. The closest one is from Summerville and Mateas [2]. They created new magic cards by means of a denoising-autoencoder. In particular, their approach allows a player to specify only part of a card's features and the model is able to fill in the blanks. Although the work of Summerville and Mateas could be applied in our scenario, evaluating mana costs is neither their end goal nor an objective in which their model is focused on.

Other works involving Magic are the ones from Ling et. al. [3] who proposes a new network type capable of code generation through the features of TCG cards and uses a dataset comprised of Magic cards to validate their results; Ward and Cowling [4] who explored the usage of bandit based Monte Carlo search applied to the problem of card selection in "Magic: The Gathering"; and Zilio, Prates and Lamb [5] that tackled the task of image-text matching.

Regarding game balancing in CCG we can highlight the work of Gold [6] which addresses the concepts of what is fun and balanced in CCGs, one of their examples being Magic. His analysis is focused on the game-play elements of a match itself. In order for a game to be fair and fun, matches need to be close and the lead needs to change. Simply adding randomness might not solve it. Another work that tackles the same problem is the one from Ham [7] which elucidates balancing challenges in games that use any sort of collectible objects. He performs several case studies and mainly focuses on the relationship between powerful cards and their rarity as well as their overall price, and how this affects a player's fun regarding game balance and the concept of what is unfair. We, on the other hand, do not feel that the rarities of cards are good balancing factors in modern CCG games. There are multiple instances of weak and cheap cards which are rare. Rather, we chose to focus on a more direct and impactful feature for balancing: their resource cost.

III. Background

In order to understand how to solve the proposed problem of predicting a resourse scale for Magic cards and comprehend the employed model, this section addresses the various concepts and methods employed in this work. The objective is to summarize the knowledge necessary to understand the techniques used. It discusses LSTM, gradient boosting, node embeddings and the SHAP algorithm.

A. Long Short-Term Memory networks

Long Short-Term Memory (LSTM) networks are an extension of Recurrent Neural Networks (RNN) that intended to remedy their problem of vanishing gradients [8]. Unlike other neural networks, the decision of a recurrent network at instant $t-1$ affects its decision at instant t. These networks receive two inputs: the present and the recent past. In their architecture, loops allow information to persist. A fraction of the network examines the segment of the input relative to instant t and returns an output. A loop feeds this output back to the network which lets the learnt information to be persisted over time. This allows the decision making at instant $t+1$ to take into account the output at instant t.

A RNN-based model often encounters the problem of vanishing gradients. Less information about the distant past is propagated at each iteration of the RNN loop. In text analysis, relationships between words that are too far apart in sentences may dissipate along time. LSTM solves this problem by storing information beyond the recent past. Data can be stored in its memory cell as well as overwritten, read or completely forgotten. Gates control how much of the memory data needs to be updated, allowing partial information propagation.

B. Node embeddings

Many important problems involving graphs require the use of learning algorithms to make predictions about nodes and edges. The main goal behind node embeddings is to map nodes to low-dimensional embeddings in such a way that similarity in the embedding space approximates similarity in the original graph. NBNE [9], [10] solves this challenge by applying a skip gram-like algorithm using nodes neighborhoods as contexts. The model learns node's representations by maximizing the log probability of predicting a node given another node within a maximum predefined distance. The main advantage of NBNE is its training speed, which is far faster than other state-of-the-art methods while still maintaining similar or better performance.

C. Gradient boosting

The main idea behind boosting is using an ensemble of weak learners that can be somehow combined to generate a stronger model. More specifically, there might be an efficient algorithm that could convert poor hypothesis, like weak learners which are slightly better than a random guesser, into a single very good hypothesis. One approach is filtering the observations, modifying the distribution of examples in such a way as to force the weak learning algorithm to focus on the harder-to-learn parts of the distribution [11].

Let y be the values of the output variable, i be an iteration of the gradient boosting algorithm and $G_i(x)$ be the output of the proposed model at time i. The algorithm improves $G_i(x)$ by constructing a new model that adds an estimator h to provide a better model, which leads to $G_{i+1}(x) = G_i(x) + h(x)$. A perfect h would imply in $h(x) = y - G_i(x)$. Therefore, the gradient boosting approach will attempt to fit h to the residual loss. However, to classification problems, residuals $y - G(x)$ for a given model are the negative gradients in respect to $G(x)$. Thus, gradient boosting is a gradient descent algorithm for combining and training weak learners. In this work we employ XGBoost, which improves upon the original gradient boosting machines [12].

D. Shapley additive explanations

Shapley value is a solution concept in cooperative game theory [13]. Let N be a set of n players in a cooperative game, S denote a coalition of players and v be a characteristic function over S. That is, $v(S)$ denotes the worth of a coalition S and describes the total expected sum of payoffs that the members of S obtain by cooperation. Adding player n_i to an existing coalition S increases the expected payoff by $v(S \cup \{n_i\}) - v(S)$. Since there are $n!$ possible ways to line up the n players and the player n_i must be preceded by all the members of S and followed by remaining players in N, there are $|S|!(n-1-|S|)!$ lineups in which player n_i joins the existing coalition S. If we sum its contribution over all lineups in which n_i joins S and over all possible existing coalitions S that it might join, we get its total contribution over all possible lineups of N. The Shapley value $\varphi_{n_i}(v)$ of player n_i is the average of its total contribution in the cooperative game (v, N)

The idea of Shapley additive explanations (SHAP) is the usage of this concept from game theory to interpret a target model [14]. We represent how model x' explains a phenomenon as a d-dimensional vector $E(x') = e_1, e_2, ..., e_d$ showing which features are contributing the model's prediction. Specifically, e_i takes a value that corresponds to the influence that the respective feature x_i had on the model decision. Since many features are vector representations of some card characteristic, we cannot assume feature independence. Correlated features end up sharing credit or importance.

IV. Proposed Approach

Our proposed approach can be divided into two separate models, one for creating a representation of a card's text and other to properly classify a card. We formulate the first model as a function $f(s; \theta_f)$ parameterized by θ_f that maps a word sequence s to a vector representation v. The second model is a function $g(v, u; \theta_g)$ parameterized by θ_g that maps a pair of vectors (v, u) to a probability distribution encompassing the classification task. Given a word sequence s, we can obtain its representation by applying $f(s; \theta_f)$ thus obtaining v. We feed v alongside the remaining features u to $g(v, u; \theta_g)$ and obtain the probabilities of each class given the inputs s and u. We employ a bidirectional LSTM to learn θ_f, XGBoost to learn θ_g and NBNE in order to reduce the dimensionality of u.

A. Embeddings of card effects

First and foremost we train word embeddings thorough the Word2Vec algorithm [15] on all words contained in the card's abilities. We filter stopwords present and perform both lemmatization and stemming over all words. Since we have diversified training instances, this becomes a necessary step to generate more robust embeddings and to avoid the occurrence of infrequent variants of relevant words.

A bidirectional LSTM layer followed by a fully connected layer are responsible for processing the input and generating a classification, in which the output of the last LSTM cell is fed to the classification layer [16]. The intuition is that this cell contains the summarized information of the whole description and is appropriate for extracting the new representation. A softmax activation over the last layer allows us to obtain the probabilities for each class. Figure 3 illustrates our proposed architecture.

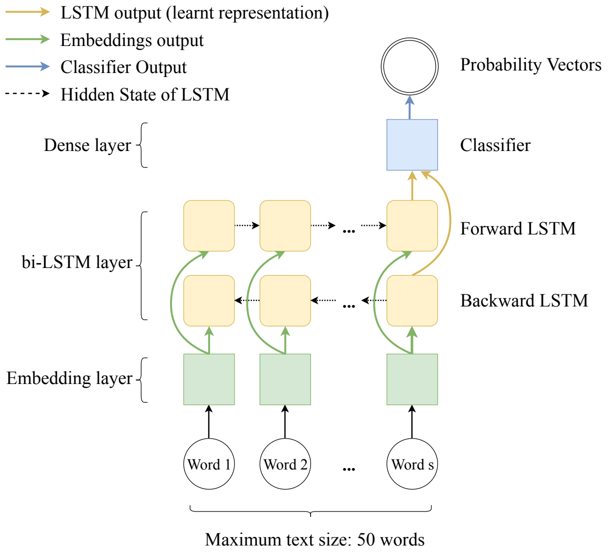


Fig. 3. Proposed neural network to perform the classification task over cards descriptions.

The architecture employed serves as a feature extraction approach and allows us obtain a new compact representation of all the textual data in a card. To obtain the final newly learned representation of the original textual features, we extract the output of the penultimate layer just before the classification step. The obtained vectors can then be used as input into any learning model and are explicitly guided by the given classes. Unlike other possible methods of extracting representations from text, this approach already introduces a bias relative to mana cost which will aid further models employed.

B. Addressing dimensionality of sparse features

Most of a card's features are composed of either categorical or list attributes. One simple solution is to hot-encode them. For features like "rarity", which has only five possible values, this approach works well. However, there are features such as printings expansions or the list of its subtypes. When hot-encoded, we get an extremely sparse representation as each one has more than 300 possible values. In fact, we obtain data of dimensionality superior to 1500 when all categorical and list features are hot-encoded. As an alternative solution, we propose to encode these features through node-embeddings.

Each of the high dimensional hot-encoded features can be modelled as a graph, where the nodes represent the cards and the edges encompass the relations inside each of the categorical classes. To illustrate this idea, let's address the graph of the printing expansions: two nodes have an edge between them if the cards they represent have at least one expansion in common. When all expansions are evaluated, we obtain an undirected graph containing all relationships between cards given their printings.

Next, we use the NBNE to obtain a compact representation of each of feature. Because it is based on the analysis of a node's neighbourhood, connected cards are close in vector space while unrelated ones end up distancing themselves. Through this approach, we are able to summarize each feature of high dimensionality. We opt to generate node-embeddings of size ten since, among the features of small dimensionality, none exceeds ten possible values. We reduce our data from over 1500 dimensions to 121, barring the representations extracted from the bidirectional LSTM.

V. Experiments

In this section, we discuss the evaluation procedures and results found associated with the proposed model, hereinafter referenced as LSTM-XGBoost. In particular, our experiments should answer the following research questions:

RQ1: In collectable card games, how well can a card's resource cost be predicted from its features?

RQ2: Does the usage of latent outputs as inputs to other models exceed a typical end-to-end network?

RQ3: What is the impact of modeling sparse categorical features as node-embeddings?

RQ4: Given collectable card games features, can we exploit some sort of inherent pattern?

RQ5: What is the impact of allowing the model to abstain from giving a doubtful prediction?

RQ6: What is the impact of specializing in the most difficult parts of the problem?

RQ1, RQ2 and RQ3 are analyzed during our general model evaluation in Section V-A. In Section V-B, we attempt to characterize data using our knowledge regarding the data while splitting the base into more concise sets, adressing RQ4 in V-B. Sections V-C and V-D are devoted to answering RQ5 and RQ6 respectively.

We consider only cards of mana cost up to 7 since cards with cost of 8 or higher are few and far between. We also filter cards with long texts (more than 50 words) or without any text at all. However, like many real world problems, the data is unbalanced. In particular, there are more cards with cost 3 than in any other class. Data augmentation is performed over the underrepresented classes by making new synthetic cards which text is compromised of random permutations of sentences from the original cards. After data augmentation, we obtain 26040 cards, of which 9039 are synthetic cards and 17001 are original ones.

To ensure the validity of the reported experiments, the original card and all its synthetic variations are always contained in the same train or test split. To ensure the relevance of the results, we assess the statistical significance of our measurements by means of a pair wise t-test [17] with $p{-}value \leq 0.05$ and through 5-fold cross validation. Unless otherwise noted, all results are statistically different from one another.

We perform an exhausting grid search to find a suitable set of hyper-parameters for both the bidirectional LSTM and XGBoost. In particular, we access the LSTM layer size (which is tied to the size of the text's vector representation) as well as the number of estimators and the depth of XGBoost trees. Our analysis leads to the usage of a bidirectional LSTM of size 150 and regarding XGBoost, 100 estimators with maximum depth of 35. We use Stochastic Gradient Descent (SGD) [18] to optimize the Cross-entropy loss function.

A. Model evaluation

Our first set of experiments address RQ1. In regards to RQ2, we use as baseline a bidirectional LSTM followed by a Multilayer-Perceptron (MLP). The core goal of the MLP is to encode and interpret the non-textual features of each card. We consider two possible architectures, one in which the MLP is employed in parallel to the LSTM and both their outputs are combined into a final prediction (LSTM+MLP) and a second one in which the outputs of the bidirectional LSTM are concatenated to the remaining inputs, and this new vector is fed to the MLP (LSTM-MLP). We also evaluate the standalone MLP, bidirectional LSTM and XGBoost models. In order to answer RQ3, we consider the scenarios with the presence of node-embeddings and without, using instead the hot encoded representations. All the results are summarized in Table I.

	Node-embeddings		*Many-hot encoded*	
	ACC	*MRR*	*ACC*	*MRR*
MLP (numeric)	.2031	.5012	*.0941*	.2634
XGBoost (numeric)	.5955	.6921	.5599	.6587
LSTM (text)	-	-	.6404	.7655
LSTM-XGBoost (text)	-	-	.6102	.7112
LSTM+MLP	.5954	.7344	.5913	.7342
LSTM-MLP	.5898	.7314	.5891	.7299
LSTM-XGBoost	**.6841**	**.8064**	**.6582**	**.7766**

TABLE I
Results obtained by the proposed models. The LSTM-XGBoost approach is statistically superior in both Accuracy and MRR.

Before addressing RQ4, we should first understand the behavior of the proposed model. We analyze the precision, recall and F_1-score for each class. Table II illustrates the performance of LSTM-XGBoost and Table III of the standalone bidirectional LSTM.

	precision	*recall*	*F1-score*	*support*
Mana 0	.97	.99	.98	3255
Mana 1	.73	.75	.74	3255
Mana 2	.47	.47	.47	3255
Mana 3	.38	.33	.36	3255
Mana 4	.46	.45	.45	3255
Mana 5	.65	.67	.66	3255
Mana 6	.82	.85	.84	3255
Mana 7	.92	.95	.94	3255

TABLE II
LSTM-XGBOOST RESULTS FOR EACH OF THE EVALUATED CLASSES. ACCURACY OF 0.6841 AND MRR OF 0.8064.

	precision	*recall*	*F1-score*	*support*
Mana 0	.93	.96	.94	3255
Mana 1	.66	.70	.68	3255
Mana 2	.42	.39	.40	3255
Mana 3	.34	.27	.30	3255
Mana 4	.42	.42	.42	3255
Mana 5	.59	.62	.60	3255
Mana 6	.78	.83	.80	3255
Mana 7	.87	.95	.91	3255

TABLE III
BIDIRECTIONAL LSTM RESULTS. THIS MODEL ONLY USES THE TEXTUAL DATA OF A CARD. ACCURACY OF 0.6404 AND MRR OF 0.7655

During further analysis around the model's confusion matrix, illustrated in Figure 4, some interesting patterns can be seen. Although the middle classes are indeed the worst performers, the model still predicts mana costs arithmetically close to their true values. This emergent behavior was not explicitly modeled and is a desirable characteristic. It serves as a strong indicator that the model does indeed learn useful general features despite the nature of the data and that it may be applied to suggesting balanced mana costs.

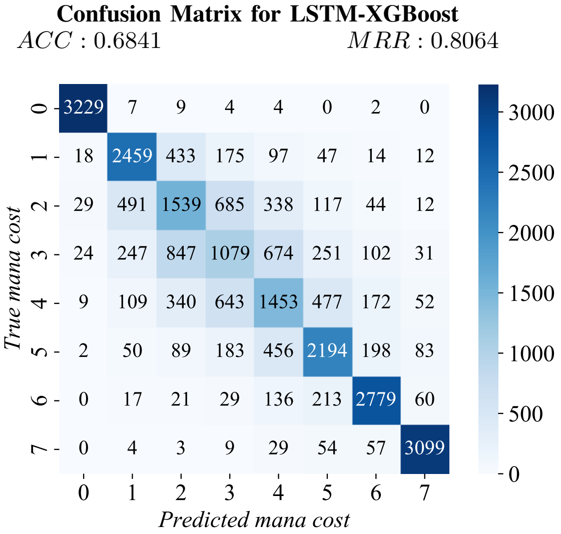


Fig. 4. LSTM-XGBoost Confusion matrix. Values are concentrated close to the main diagonal, implying that the model learned useful rules.

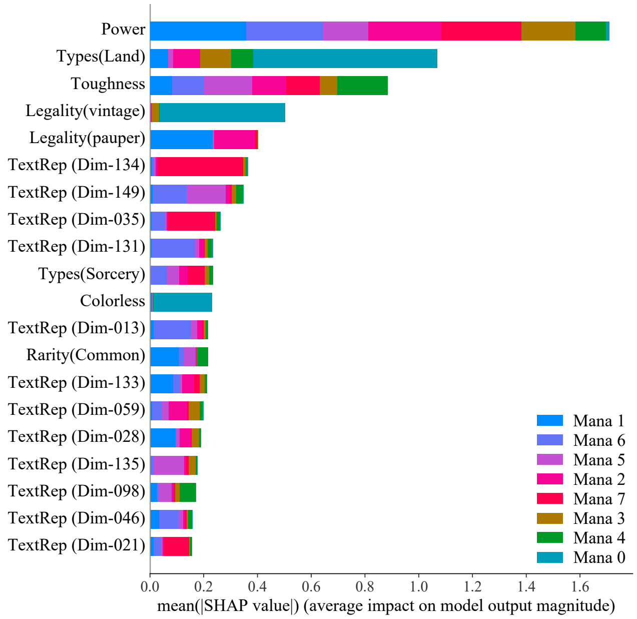


Fig. 5. SHAP values of the top 20 features used for classification. Several dimensions related to text representation appear as top performers.

We employ the SHAP algorithm to visualize the impact of each feature used as input to LSTM-XGBoost. It is possible to observe that a considerable part of the model's explanation is a result of a card's abilities representation. Among the twenty most relevant features, twelve of them are some dimension of the text vector representation as seen in Figure 5. Regarding the numerical attributes, there is not much that can be done in order to improve the model besides experimenting with new modeling forms. However, if we were able to generate more robust representations of each card's text in such a way that they better classify the data, we should observe an improvement in the predictions. The next experiments involve mainly LSTM and how to address this task. We chose to analyze the LSTM individually rather than the full LSTM-XGBoost model to directly access the impact of our decisions over the representations, explicitly filtering any correlation that might exist with the remaining features.

B. Data characterization

A common phenomenon in games published over a long period of time and which allow the usage of both old elements and new ones is power creep [19]. The idea behind the concept is that a company has to sell their new products, but everything new they create has to compete with previously existing pieces. To draw consumers' attention and justify further acquisitions, new products end up becoming superior to older ones to the point of becoming strictly better and completely outclassing one another. This means that older content becomes obsolete or relatively under-powered. In "Magic: The Gathering", in each new edition developers attempt to power up some aspects of new cards while bringing down in power other ones, overall balancing things out.

	93-00	01-06	07-11	12-14	15-16	17-19
Mana 0	.93	.94	.97	.94	.97	.98
Mana 1	.72	.69	.65	.68	.63	.67
Mana 2	.43	.38	.38	.39	.35	.41
Mana 3	.32	.30	.26	.26	.30	.29
Mana 4	.45	.44	.37	.37	.38	.40
Mana 5	.62	.62	.57	.65	.62	.66
Mana 6	.84	.82	.81	.84	.84	.83
Mana 7	.87	.94	.92	.92	.93	.94
Accuracy	.61	.65	.62	.66	.65	**.68**
MRR	.75	.76	.75	.77	.77	**.79**

TABLE IV
CLASS F_1-SCORE, ACCURACY AND MRR FOR EACH TIME SPAN.

	🔥	🌳	💧	💀	☀	◇	❂
Mana 0	.97	.96	.97	.96	.98	.95	.99
Mana 1	.67	.63	.65	.66	.69	.66	.66
Mana 2	.34	.32	.38	.36	.42	.43	.35
Mana 3	.29	.29	.32	.30	.34	.29	.42
Mana 4	.36	.47	.41	.41	.45	.45	.39
Mana 5	.59	.59	.60	.63	.64	.66	.65
Mana 6	.81	.80	.79	.78	.81	.87	.81
Mana 7	.91	.93	.90	.94	.89	.93	.93
Accuracy	.62	.63	.62	.62	.64	.70	.78
MRR	.75	.76	.75	.75	.76	.80	.85

TABLE V
CLASS F_1-SCORE, ACCURACY AND MRR FOR EACH COLOR TYPE.

Of course, given the rather large span of time and the high number of cards already printed, it becomes unfeasible to promote a globalized human-based balancing approach. This leads to developers mainly focusing on the recent past and in popular traditional cards, which in turn leads to some degree of power creep. Over the nearly three decades of its existence, many of the design decisions have changed. New abilities were introduced, new combinations of cards emerged and the game became more dynamic. With this in mind, we propose an analysis of Magic in intervals of time as shown in Table IV. Each time split contains roughly the same amount of cards. This leads to a far more reliable and concise analysis, especially because the card distribution over the years is rather different.

The color system is one of the game's most fundamental and iconic elements. It gives the game diversity in its cards, effects, and play styles, while preventing any one deck from having every tool in the game. Each of the five colors represents a set of beliefs and principles [20]. A color's philosophy explains how it sees the world, what objectives it hopes to realize and what resources and tactics a color has at its disposal. Gameplay-wise, this dictates which card types and abilities thematically fit within a color. With this in mind, we explore creating different models for each of core Magic's colors as well as one for colorless cards and multi-colored ones.

"Magic: The Gathering 2011 player's handbook" provides an overview of each color. White ☀ comes from plains, meadows and fields, they bring light and order. Blue 💧 comes from islands and bodies of water, involving intellect and illusion. Black 💀 comes from swamps and places of death. Black magic is steeped in darkness and death. Red 🔥 comes from mountains and rocky places and call forth fire and passion. Green 🌳 comes from forests and jungles and conjure growth and might. Colorless ◇ are unbound and can come from a variety of places. Multicolored ❂ are flexible and encompass all the philosophies from each of their colors.

We assume a learning scenario in which cards can be mapped to context domains, associated with their color identities, enabling us to learn specific models for each color domain. Our main hypothesis is that there are abilities that might be more valuable in certain domains than in others. This should be reflected by Magic's design choice of associating each color with a philosophy and a mechanic. For instance, if an ability is vastly present in some color but scarce in others, it might be the case that it should be valued differently in each context. Further, cards that have more than a single color or no color at all are more flexible regarding their abilities and strategies. It should be useful to address them separately from other cards. Table V summarizes the experiment regarding these assumptions.

C. Dealing with uncertainty

The output of LSTM-XGBoost is composed of a probability distribution which indicates the most likely classes for each of the input instances. From information theory, the more equally distributed a probability distribution, the greater its entropy and therefore the greater the uncertainty over the prediction of the modeled random variable [21]. During training, the models are optimized with the cross-entropy loss function in an attempt to minimize uncertainty of the model's output by maximizing the likelihood of predicting the correct classes while minimizing the likelihood of incorrect ones. Even so, there are cases where the model presents considerable uncertainty due to the design setting of superficially identical cards having different mana costs.

We propose establishing a certainty threshold during the evaluation step as the minimum probability that should be associated with a single class to validate the proposed prediction in RQ5. Indirectly, this modeling entails classifying low entropy instances and abstaining from high entropy cases. Under these constraints, the model presents a remarkable gain in performance as shown in Figure 6 while still classifying over half the validation instances.

D. Domain adaptation on hard classes

In order to answer RQ6, we explore a transfer learning approach related to our problem. In particular, we first train the model using all data samples and then perform domain adaptation over the intermediary classes where the model usually struggles. Table VI illustrates the new results in the specialized classes, and an improvement can be seen. Table VII shows the model performance including the remaining classes. Although the performance for the specific specialized classes improves, the model forgets how to handle the easier classes and ends up with an overall worse performance. This leads us to believe that domain adaptation in the most difficult classes is not a suitable approach.

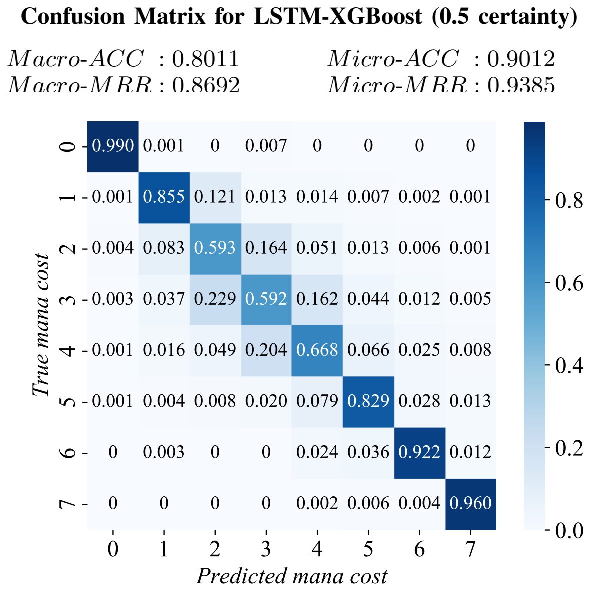


Fig. 6. LSTM-XGBoost Confusion matrix setting a certainty threshold of 0.5. Since under this constraint each class presents a distinct amount of instances, we opt to use the percentage of occurrences rather than their absolute values for better visualization.

	precision	*recall*	*F1-score*	*support*
Mana 2	.51	.48	.49	3255
Mana 3	.38	.36	.37	3255
Mana 4	.45	.48	.47	3255
Mana 5	.64	.67	.66	3255

TABLE VI
BIDIRECTIONAL LSTM RESULTS AFTER DOMAIN ADAPTATION IN THE MOST DIFFICULT CLASSES.

	precision	*recall*	*F1-score*	*support*
Mana 0	1.00	.11	.20	3255
Mana 1	.00	.00	.00	3255
Mana 2	.24	.49	.33	3255
Mana 3	.20	.37	.26	3255
Mana 4	.24	.48	.32	3255
Mana 5	.33	.68	.44	3255
Mana 6	1.00	.00	.00	3255
Mana 7	.98	.03	.05	3255

TABLE VII
BIDIRECTIONAL LSTM RESULTS AFTER DOMAIN ADAPTATION IN ORDER TO SPECIALIZE IN THE MOST DIFFICULT CLASSES. THE MODEL FORGETS HOW TO HANDLE THE ORIGINALLY EASY CLASSES AND HAS AN OVERALL WORSE PERFORMANCE. ACCURACY OF 0.2693 AND MRR OF 0.4874.

E. Results and discussion

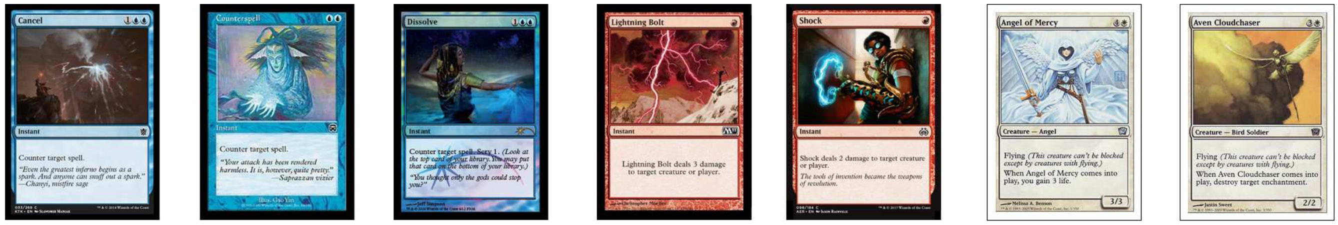
Among the first set of experiments, we can visualize the impact of modeling sparse features as node-embeddings. We can see an improvement in performance in all cases barred the LSTM combined with MLP scenario. They are indeed useful even though no node feature appear amongst the top performer features. The data's dimensionality reduction allows the model to properly focus on more relevant features while still pertaining information regarding the sparse ones.

Overall, the developed models present high performance evidenced by their MRR. A desirable emergent behavior is that when the model misclassifies the cost of some card, it's usually attributed to an arithmetically close mana cost. This was verified during our analysis over the confusion matrix which presents high values close to the main diagonal. It leads us to believe that the LSTM-XGBoost does indeed learn general useful features for a resource scale.

Other arguments corroborating with the proposed hypothesis emerge from the SHAP analysis. We can see a strong influence of features like power or toughness of a card, which was expected, but we also see the presence of features related to the text representation of a card. In particular, we observe that some dimensions are more closely associated with some classes than others. A couple examples include dimension 134 which is extremely important to cards with cost of 7 or dimensions 13 and 131 which present strong ties with cards of cost 6. We can also identify some dimensions that appear to be useful to differentiate between cards of close mana cost, like dimension 98 having considerable impact involving costs 4 and 5 or dimension 133 which is useful to costs 1 and 2.

Although we cannot explicitly state what each dimension represents, it's clear that there are ties between specific mana costs and certain characteristics present in the text of each card. The model learns to make these connections and provide mana costs scale recommendations given the card's features. Take for instance the cards shown in Figure 7. *Cancel* and *Counter spell* have same exact text but distinct mana costs. Likewise, *Lightning bolt* and *Shock* have the same mana cost but one does slightly more than the previous one. Our model returns that *Lighting Bolt* should have a cost of two and *Shock* is appropriate as is. Although the model miss classify the blue cards, it provides consistent results, classifying the two equal cards similarly and the stronger one is attributed a higher cost. It is also able to understand the differences in similar cards and correctly predict which is stronger, as shown with *Angel of Mercy* and *Aven Cloudchaser*. Not only it's able to capture what similar cards have in common, but also what they have that sets them apart. Cards with intermediary mana costs compromise the majority of published cards as well as the most common cards found in player's decks. In a certain way it's natural that they are the ones with the largest variance.

A useful way to attempt to deal with this is to split the data into more concise bins. We explore making this division based on either release date or their color. We can observe that the model trained upon the recent years is the performant while the one trained on the oldest cards is the worst performer. Many instances from cards with similar effects and divergent mana costs arise from new cards being compared to old ones. We could take advantage of this phenomenon and deploy models based primarily on recent years. Regarding colors, we do not see any improvement when dealing with regular colored cards. However, we observe a great increase in performance when dealing with both colorless and multicolored cards. This seems to imply that cards from these two scenarios are significantly different from the other ones. Further analysis is required, but it might be the case that we should also focus on cards that are exclusively of a color rather than simply considering if they have that color in their identity.



Predicted mana cost: 1 Predicted mana cost: 1 Predicted mana cost: 2 Predicted mana cost: 2 Predicted mana cost: 1 Predicted mana cost: 5 Predicted mana cost: 4

Fig. 7. Some examples of “Magic: The Gathering” cards and their predicted mana costs.

By far, the largest gain in performance is when we allow the model to abstain from giving a prediction. Most cases in which the model performs a miss classification arise from the inherent ambiguity of the input. When dealing only with instances that the model has moderate certainty, we reach an MRR of 0.9385. Since some classes are easier than others, the model naturally gives more predictions of these classes. Considering macro MRR, we still maintain a good performance of 0.8692. This serves as yet another argument in corroboration of our hypothesis and in favour of the proposed approach.

VI. Conclusion

In this work we proposed a novel approach to dealing with the task of recommending mana costs for “Magic: The Gathering” cards. To our knowledge, this has never been done before and, as such, we propose an in-depth analysis of the peculiarities of the task at hand. Under usual circumstances, the proposed model reaches a MRR of 0.8064 but we show that, given some restrictions, this result can be improved upon.

Through our experiments, we present several arguments that corroborate with the hypothesis that it is indeed possible to learn useful general features that explain a card’s mana cost. The instances that the model miss classify the inputs might not be “true errors” given the ambiguity of mana costs and how cards exist in the wild. It could be the case that these cards are indeed unbalanced and the model is instead proposing a suitable new mana cost for them. In order to further validate our model, a simple yet effective method would be a qualitative evaluation of its output. One approach could be the creation of new synthetic cards which have random attributes extracted from our database. Given the model’s output for these new cards, we could ask the opinion of seasoned Magic players regarding their mana cost and their thoughts regarding its balance.

Aside from creating models for each studied scenario, some directions for future work also include the usage of Generative Adversarial Networks (GAN). In order to augment the under-represented classes we simply created new permutations of a card’s text. The main issue with this approach is that these classes remain with a considerably smaller vocabulary. Using GANs to perform data augmentation would allow us to have a broader vocabulary for all considered classes.

Acknowledgements

This work was partially funded by the authors individual grants from CNPq and FAPEMIG.

References

[1] R. Garfield, “Magic: The gathering,” *Game {Board Game}.(1993). Wizards of the Coast, Renton, Washington, US*, 1993.

[2] A. Summerville and M. Mateas, “Mystical tutor: A magic: The gathering design assistant via denoising sequence-to-sequence learning,” in *12th Artificial Intelligence and Interactive Digital Entertainment Conference*, 2016.

[3] W. Ling, P. Blunsom, E. Grefenstette, K. Hermann, T. Kočiský, F. Wang, and A. Senior, “Latent predictor networks for code generation,” in *54th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2016, pp. 599–609.

[4] C. Ward and P. Cowling, “Monte carlo search applied to card selection in magic: The gathering,” in *IEEE Symposium on Computational Intelligence and Games*, 2009, pp. 9–16.

[5] F. Zilio, M. Prates, and L. Lamb, “Neural networks models for analyzing magic: The gathering cards,” in *25th International Conference on Neural Information Processing, ICONIP*, 2018, pp. 227–239.

[6] K. Gold, “Why games must be more than fair: Random walks, long leads, and tools to encourage close games,” 2010.

[7] E. Ham, “Rarity and power: balance in collectible object games,” *Game Studies*, vol. 10, 2010.

[8] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[9] T. Pimentel, A. Veloso, and N. Ziviani, “Unsupervised and scalable algorithm for learning node representations,” *International Conference on Learning Representations*, 2017.

[10] T. Pimentel, A. A. Veloso, and N. Ziviani, “Fast node embeddings: Learning ego-centric representations,” in *6th International Conference on Learning Representations*, 2018.

[11] R. Schapire, “The strength of weak learnability,” *Machine learning*, vol. 5, no. 2, pp. 197–227, 1990.

[12] T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system,” in *22nd SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794.

[13] L. S. Shapley, “A value for n-person games,” *Contributions to the Theory of Games*, vol. 2, no. 28, pp. 307–317, 1953.

[14] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” in *Advances in Neural Information Processing Systems*, 2017, pp. 4765–4774.

[15] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *Advances in Neural Information Processing Systems*, 2013, pp. 3111–3119.

[16] G. L. Zuin, L. Chaimowicz, and A. Veloso, “Learning transferable features for open-domain question answering,” in *2018 International Joint Conference on Neural Networks, IJCNN*, 2018, pp. 1–8.

[17] T. Sakai, “Statistical reform in information retrieval?” *SIGIR Forum*, vol. 48, no. 1, pp. 3–12, 2014.

[18] L. Bottou, “Large-scale machine learning with stochastic gradient descent,” in *International Conference on Computational Statistics*, 2010, pp. 177–186.

[19] T. Falcão and D. Marques, “Pagando para vencer, parte 2: Serialização, power creep e capitalismo tardio em hearthstone,” in *41° Congresso Brasileiro de Ciências da Comunicação*. Intercom, 2018.

[20] M. Rosewater, “Pie fights.” Wizards of the Coast, Renton, Washington, US, 2016, accessed on 22-March-2019. [Online]. Available: https://magic.wizards.com/en/articles/archive/making-magic/pie-fights-2016-11-14

[21] C. E. Shannon, “A mathematical theory of communication,” *Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.