Discussion/solution.

Without the cardinality condition, there is a unique solution x only if A has zero nullspace. This requires that $m \geq n$ and that A have rank n. When we add the cardinality information, it can happen that we have a unique solution, even when $m < n$, or $\mathbf{Rank}(A) < n$. These ideas are central to a new research area called compressed sensing. But back to our problem ...

Let's consider *any* set of k indices. Form the matrix $\tilde{A} \in \mathbf{R}^{m \times k}$, taking only the associated columns of A. Now consider the equation $\tilde{A}z = y$. Any solution of this equation gives us a solution x of $Ax = y$, with $\mathbf{card}(x) \leq k$, just by inserting the entries of z into the positions of x associated with the indices, with zeros elsewhere. If the equation $\tilde{A}z = y$ has more than one solution, then the original x is not recoverable; there are at least two values of x that satisfy $Ax = y$ and $\mathbf{card}(x) \leq k$ (indeed, the two solutions have the same sparsity pattern). So the equation $\tilde{A}x = y$ can have only one or zero solutions. If $\tilde{A}x = y$ has one solution, then it is for sure a candidate for x.

Now, we carry out this analysis of the equation $\tilde{A}z = y$ for *all* $\begin{pmatrix} n \\ k \end{pmatrix}$ choices of k indices from $1, \ldots, n$. If for any choice of indices there is more than one solution, we can't recover x. We can just quit the whole process right there.

If for all choices that have a solution, the solution is the same, then that vector is x, and it is the unique solution.

There are several ways to carry out this method. (There are also several incorrect ways to do it.) Here is one correct way: For each subset, check if $\tilde{A}z = y$ has a solution. If not, go on to the next subset. If it does, check the rank of $\tilde{A}$. If it is less than k, quit the entire algorithm, announcing

Now, this isn't really practical, since $\begin{pmatrix} n \\ k \end{pmatrix}$ is a really big number, unless k is very small. But I didn't ask for a practical method.

None of the following was needed, but you might find it interesting. It is likely there isn't a much better way to answer the question with certainty than to do an exhaustive search over subset of cardinality k. However, there are some very good heuristics for finding a sparse x that satisfies $Ax = y$. One way is to minimize $\|x\|_1$ subject to $y = Ax$. This can be done using linear programming. This is a heuristic — it can be wrong — but it very often does recover a sparse x from $y = Ax$.